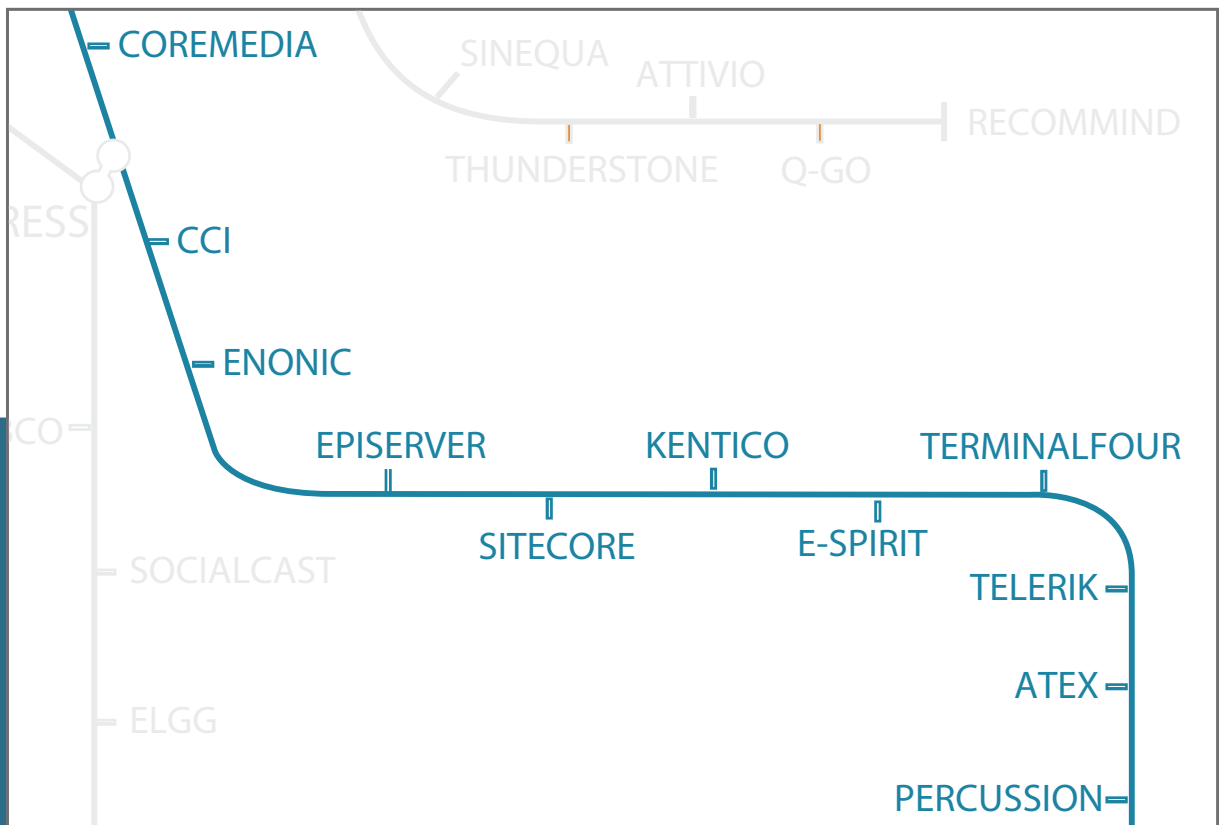


Real Story

MAKE BETTER
TECHNOLOGY DECISIONS

GROUP



The Web Content & Experience Management Evaluation Report

Version 22.3

Comprehensive Product Evaluations

Background & Acknowledgments

About Your License

You have purchased the Enterprise Edition of the report. This publication may not be copied, reproduced, or retransmitted in any form or by any means, *including storage and retrieval system*, without prior written consent.

Copyrights

Real Story Group and the Real Story Group logo are trademarks of CMSWorks, Inc., All other company names, product names, and service marks are used for identification purposes only and are trademarks of their respective owners. Except where noted, all screenshots came from product vendors themselves. All other graphics are copy written by CMSWorks, Inc., and may not be copied, reproduced, or reused without express written permission.

Note to Vendors: No Commercial Use Policy

Real Story Group accepts no advertising from and is not beholden to any supplier listed in any of our reports. Our income is derived from the sale of this and other publications as well as consulting for product buyers only. Our analysis, reports, and information are intended solely for the use of our readers. *Neither the information in this report nor the name of Real Story Group or any of its publications, may be used in advertising or for any other commercial purpose, including any use on the Internet or mention in any press releases or newsletters in print or electronic form.* Real Story Group will take all steps open to it to prevent commercial use of its materials and its name.

About the Authors

Several current and former Real Story Group analysts combined to deliver this report, including Apoorv Durga, Sarah Brown, Adriaan Bloem, Irina Guseva, and Tony Byrne. For more information, please contact The Real Story Group at feedback@realstorygroup.com, or www.realstorygroup.com.

Acknowledgments

Special thanks to Lyza Morss for editing and production.

Over the years, hundreds of Web CMS customers shared their experiences with specific tools. As a firm rule, we do not list their names, but we appreciate their input, which underpins the bulk of this report.

Important Disclaimers

The product-specific information contained in this document is intended to provide an overview of a specific product and vendor at the date of publishing. CMSWorks, Inc. cannot ensure the accuracy of this information since products, vendors, and market conditions change rapidly. The authors and CMSWorks, Inc. disclaim all warranties as to the accuracy, completeness, or adequacy of the information in this report and shall have no liability for errors, omissions, or inadequacies in the information presented. A complete assessment of your specific application, the method of implementation for a given product or technology,

and the current state of that product in your specific environment must be considered in order for a recommendation to be made regarding any product's suitability for your purposes and needs.

Your Feedback and Questions

Have any feedback or questions about this report? Write to feedback@realstorygroup.com.

Executive Summary	1
The Case for Web Content and Experience Management	1
Web Content and Experience Management Technology	2
The WCXM Marketplace and Vendors	4
Advice, Pitfalls, and Best Practices	6
Part 1 – How to Use this Report	8
Brief Outline	8
Browse by Role	8
Part 2 – Content, Experience, Management, and Systems	10
The Digital Content Mandate: Captivate the Visitor	10
Larger and More Complex Websites	10
Content and Customer Experience Management	11
More Demands	11
Publish Faster	12
Publish Cheaper, Reuse More	12
Support Social Participation, Increase Customer/Employee Satisfaction	13
What Is a Content Management System (and What Is It Not)?	13
The Business Case: How to Justify a WCM Investment	13
Hard Benefits	14
Soft Benefits	15
Which Rationale?	16
When All Else Fails	17
Part 3 – CMS Dissected	19
Content Management Basics	19
Portals	20
Digital Asset Management (DAM)	20
Digital Marketing	21
Web Content & Experience Management (WCXM)	21
How CMS Tools Work	22
Technology Services	23
Standards and Guidelines: Aligning with the Rest of the Industry	23
Technical Administration and Security	33
Development Services: Making Your CMS Your Own	45
Performance: Speed and Reliability Matter	53
Content Services	57
Contributor Experience: How Content Managers Use the System	57
Contributing Content	64
Content Lifecycle Services	78
Publishing Services: Content Delivery	89
Digital Marketing Services: Engagement, Measurement, and Optimization	95
Workplace Services: Supporting the Employee Digital Experience	107

Ancillary Services: Beyond Content Management and Social Apps	108
Online Forms: Many Uses for Experience Management	110
Module Ecosystem: Leveraging Additional Functionality	111
Vendor Intangibles Dissected	112
Vendor Services	113
Universal Scenarios: The Key to Comparing Technologies	116
Simpler Site	118
Mid-Range Site	119
Complex Site	122
Digital Workplace	126
Part 4 – CMS Product Comparisons	129
Overview of the Landscape: Growth and Some Specialization	129
Assessing Different Delivery and Licensing Models	129
The Hosted, “Software as a Service” Model	129
Community Open Source	131
Commercial Open Source	132
Examining Basic Specifications	133
How We Categorize Vendors	133
Product Categories by Tier	133
Matching Features to Requirements	137
Putting Scenarios and Vendors Together	139
How to Use All of these Charts	142
Legacy Platform Vendors: Roll-Up Comparisons	143
HP: TeamSite	151
IBM: Web Content Manager	174
OpenText: Web Experience Management	195
Oracle: WebCenter Sites	219
Upper-Range Platform Vendors: Roll-Up Comparisons	239
Adobe: AEM Sites	248
CoreMedia: CMS	273
Percussion: CM System	286
SDL: SDL Tridion	307
Sitecore: Sitecore CMS	330
Mid-Range Platform Vendors: Roll-Up Comparisons	354
Atex: Polopoly	363
Drupal: Drupal	372
EPiServer: EPiServer CMS	386
Hippo B.V.: Hippo CMS	407
Microsoft: SharePoint Server 2013	423
Plone: Plone	443
TYPO3: TYPO3	460

Mid-Range Product Vendors: Roll-Up Comparisons	476
CrownPeak: CrownPeak CMS	490
Ektron: Ektron CMS	508
Enonic: CMS	528
e-Spirit: FirstSpirit	540
eZ Systems: eZ Publish Platform	557
Hannon Hill: Cascade Server	578
Ingeniux: Content Management System	592
Upland Software: Clickability	612
Magnolia: Magnolia	633
CCI: Escenic	648
Simpler Product Vendors: Roll-Up Comparisons	662
DotNetNuke: DotNetNuke	673
Joomla!: Joomla!	693
Kentico: Kentico CMS	709
MODX: Revolution	730
OmniUpdate: OU Campus	747
OpenCms: OpenCms	764
Salesforce: site.com	778
Telerik: Sitefinity	789
TERMINALFOUR: Site Manager	801
Automatic, Inc.: WordPress	815
Do It Yourself?	833
Regional and Niche Players	834
Part 5 – Advice, Pitfalls, and Best Practices	836
How to Select a Web CMS Package	836
Vendor Pricing	843
Software	844
Professional Services and Support	845
Some Final Words on Pricing	846
Twelve Common Pitfalls to Avoid (and Best Practices to Follow)	847
Glossary	854
Vendor Index	862

Executive Summary

Welcome to the *Web Content and Experience Management (WCXM) Report*. The next several pages try to distill down this long report for senior managers and others with short attention spans. You may want to employ this — and the next section “Part 1 – How to Use this Report”— to decide where to read more deeply.

The Case for Web Content and Experience Management

Content is king, but visitors on your websites have higher expectations than just being able to read an article about one of your inventions, viewing your calendar of events, or buying a product that you sell. The websites are growing larger, more diverse, and more functional. At the same time, the digital audience is getting more impatient and expectant of high-quality standards across all of your channels: mobile, web, email, and social.

Web is one of the most crucial channels for modern organizations, and large enterprises are particularly focused in their efforts to provide optimal experiences for their audiences by appropriately using web content management systems in combination with digital marketing, social analytics, and similar tools. Responsive design and template management aside, a WCM system is also a vehicle for multilingual content management and globalization. A web content management system is often the central repository for the management and delivery of content translated into multiple languages, whether it comes from a translation management system or it is manually handled by human translators.

Depending on the specific WCM system, it is possible to manage mobile delivery and associated asset renditions and layouts from the same platform. That is, if a prospect is looking at an item she is interested in purchasing, she can have an uninterrupted experience as she moves from the web interface to continue her journey on a tablet, to possibly finalizing the transaction from her mobile phone: anywhere, at any time, on any device.

Since the inception of web content management technology in the early 1990s, this field has seen dramatic evolution. Through iterations of change from nomenclature, to features and functionalities, to acronyms, to marketing branding and messaging — both the WCM vendor-scape and organizations focused on managing their digital content have contributed to the evolution of WCM. This has expanded the notion of WCM from mere content management on the web to also managing the experience associated with digital content that is consumed by your audiences.

There are many benefits to employing a Web CMS or “Web Content and Experience Management” (WCXM) technology. When done right, it can yield various benefits, both hard and soft.

Hard

- Increase transactions
- Accelerate time to market
- Achieve process efficiencies
- Increase customer acquisition and retention rates
- Increase customer satisfaction and support

Soft

- Put business people in control of digital communications
- Maintain consistent user experience and brand
- Improve agility
- Strengthen security
- Innovate and improve a competitive stance
- Maximize internal resources

Oftentimes, however, these benefits cannot be readily translated into a clear “Return on Investment,” and as such, you may need to resort to a “Cost-of-doing-business” justification.

Web Content and Experience Management Technology

Web content management is related to (but distinct from) other content management tools and disciplines, including Enterprise Content Management (ECM). ECM primarily concerns itself with processing information; web content management primarily concerns itself with publishing information on the web, but also across other channels such as mobile.

Functionality

We analyze web content and experience management tools across three different sets of services:

1. **Technology Services**, including development and administrative services, as well as performance and scalability
2. **Content Services**, including UI usability, taxonomy management, workflow, and globalization
3. **Experience Services**, including site & campaign analytics, mobile, and social media integration, among others

To some extent, Content Production and Delivery could be considered together as a Web CMS functional lifecycle, but they are architecturally distinct environments, and not all products cover both ends of the spectrum.

By understanding your specific needs and opportunities in each dimension, you can begin to construct the outlines of a new CMS. You will also have a roadmap for evaluating the suitability of various package offerings.

Dimensions of Web Content and Experience Management		
Technology Services	Content Services	Experience Services
<p>Technical Administration & Security</p> <ul style="list-style-type: none"> • Threat Prevalence • Authentication & Authorization • System Reporting • Multisite Management • Cloud Services <p>Development</p> <ul style="list-style-type: none"> • Configuration & Customization • Integration & Extension • Content Modeling • Templating <p>Performance</p> <ul style="list-style-type: none"> • Back-end Performance • Site Caching & Delivery 	<p>Contributor Experience</p> <ul style="list-style-type: none"> • Overall Usability • UI Accessibility <p>Contributing Content</p> <ul style="list-style-type: none"> • Authoring & Transformation • Tagging & Taxonomy • Content Reuse • Media & Document Management • Repository Services <p>Content Lifecycle</p> <ul style="list-style-type: none"> • Workflow • Globalization • Archiving & Compliance 	<p>Publishing</p> <ul style="list-style-type: none"> • Standards Adherence • Multichannel • Mobile <p>Digital Marketing</p> <ul style="list-style-type: none"> • Site & Campaign Analytics • Testing & Optimization • Personalization & Segmentation • Social Media Integration • Promotional Campaigns • Community & UCG <p>Workplace</p> <ul style="list-style-type: none"> • Collaboration & Networking • Dashboard <p>Ancillary</p> <ul style="list-style-type: none"> • Site Search • Online Forms • Module Ecosystem

Usage Scenarios

Explicitly or not, different CMS products target different use cases. Understanding the business scenarios that fit better or worse for the different packages enables you to see deeper into their relative strengths and weaknesses *for your particular circumstances*. Therefore, we have identified 11 common scenarios against which vendors can be judged. They are useful for understanding what types of products tend to work better for each type of project.

Simpler Site

- Informational
- Microsites & Landing Pages

Mid-Range

- Basic Digital Marketing
- Mobile Site
- Community Oriented

Complex

- Advanced Marketing Portal
- Global Enterprise
- Multichannel Publishing
- Ultra-Large Single

Digital Workplace

- SMB/Departmental
- Global/Enterprise

The WCXM Marketplace and Vendors

We break the marketplace up into several tiers, roughly from more complex/expensive to simpler/cheaper. For each vendor below, you can find a 12-20 page evaluation later in this report, along with several charts comparing all the peers in each tier.

Legacy Platforms

- **HP** – Autonomy TeamSite is an old and somewhat faded grand dame of Web CMS, and is much friendlier for authors than developers
- **IBM** – Comparatively obscure offering living a bit uneasily in the company's WebSphere group (albeit with decent integration to WebSphere Portal Server)
- **OpenText** – Arguably, the first serious CMS tool (former Vignette) remains a very complicated and expensive platform
- **Oracle** – Long-standing (if somewhat creaky) product (former FatWire), which focuses on personalization and catalog management

Upper-range Platforms

- **Adobe Experience Manager Sites** – Under Adobe, AEM Sites is constantly evolving, but it remains a difficult platform to master for businesspeople and technologists alike
- **CoreMedia** – German vendor with a strong footprint in the media sector that is struggling a bit in North America
- **Percussion** – Slow-footed vendor traditionally focused on supporting structured content for multichannel publishing that has shifted its focus down-market to its much simpler product called CM1
- **SDL Tridion** – British/Dutch vendor compensates for dated codebase, with advanced facilities for multisite, multinational publishing scenarios
- **Sitecore** – Danish vendor expanding around the globe with a quite complicated, .NET-based platform that appeals mightily to its consulting partners

Mid-range Platforms

- **Atex Polopoly** – UK-based, J2EE offering that combines various Java standards and popular components (such as Lucene/Solr) with a proprietary architecture

- **Drupal** – Everybody’s favorite development platform actually has serious shortcomings in the enterprise
- **EPiServer** – Sweden-based vendor promotes modern, .NET-based solution to compete with Sitecore and Ektron
- **Hippo** – Java-based platform focused on component-oriented publishing, still with its biggest footprint in Benelux
- **Microsoft** – Incorporates Web CMS features into SharePoint with rather disappointing results
- **Plone** – One of the most longstanding open source CMS platforms, which appeals to developers with a good command of Python and Zope
- **TYPO3** – Longstanding, PHP-based, community open source platform that has been fading away in the past years — particularly outside of its home base in Europe

Mid-range Products

- **CrownPeak** – A historically slick, SaaS-based offering that has faltered a bit in the marketplace
- **Ektron** – One of the oldest .NET WCM vendors, the firm and offering have lacked a bit in consistency over the years
- **Enonic** – Norway-based vendor offering low-cost alternative, with a fairly recent focus on web experience
- **E-Spirit** – Germany-based solution somewhat longer on engineering than usability
- **eZ Systems** – Norway-based, commercial, open source offering that tries to compete with Drupal, despite its somewhat arcane, PHP-based code
- **Hannon Hill** – US vendor promotes its XML capabilities, with special attention to the higher-education sector
- **Ingeniux** – US vendor with somewhat dated technology but strong knowledge of higher education
- **Upland Software (Clickability)** – Longstanding, SaaS-based platform that keeps finding new owners and has slipped into comfortable obscurity
- **Magnolia** – Commercial open source platform based on Java is ideally suited to larger, page- and folder-oriented publishing regimes
- **CCI Escenic (formerly Vizrt)** – Newspaper-oriented system has seen multiple owners and consequently lost substantial R&D momentum

Simpler Products

- **DotNetNuke** – Somewhat aging and bulbous .NET offering with broad, raucous support ecosystem; more portal-like than CMS-oriented
- **Joomla!** – A nice, PHP-based system for smaller websites; avoid in security-conscious settings
- **Kentico** – Czech-based Kentico is a .NET, page-based system
- **MODX** – The Texas-based company maintains two open source WCXM projects; a legacy product “Evolution,” and its recent successor, “Revolution”
- **OmniUpdate** – Long-standing player targets higher education and local government with comparatively lower price structure
- **OpenCMS** – Longstanding, Java-based offering centered in the DACH region is not overly complicated, and is best suited for simpler sites
- **Salesforce** – SaaS-based site.com is an extremely lightweight but rapidly evolving entry-level WCXM offering for existing customers of its popular CRM platform
- **Telerik** – Bulgarian firm with aggressive North American expansion plans targets Ektron/Sitcore customers with lower price point
- **TERMINALFOUR** – Irish/UK vendor that emphasizes accessibility and higher-ed credentials
- **Automattic WordPress** – Enormously popular blogging platform is spreading its wings to WCM systems, with a simple setup and ease of use

Advice, Pitfalls, and Best Practices

Most CMS customers who end up disappointed can trace their problem to an inadequate and/or unstructured technology selection process. It’s essential to follow a methodical review process based on actual system and usability testing, rather than long lists of check-box requirements.

Throughout this process, you want to assess the vendor as much (or more) as you would the product. Key areas of exploration here are:

- Vendor Professional Services
- Channel Partner Services
- Support & Community
- Strategy & Roadmap
- Viability & Stability

Web CMS projects (like all IT endeavors) can easily fail. It’s essential to follow the lessons of others who have gone before you. Among the dozen best practices we identify, these stand out:

- Obtain strategic direction, a suitable budget, and a mandate for necessary changes
- Involve system users in the selection, design, implementation, and testing of the system
- Understand the different roles and motivations of consultancies, integrators, and vendors

- Anticipate future expansion in both your content and technical architectures
- Do not forget about content strategy and digital marketing in your content management activities

Part 1 – How to Use this Report

This is a very long report. That’s intentional. You’re going to make some important decisions on behalf of your enterprise, and our goal is to provide you with as much relevant detail as possible. However, we know you may not have time to read cover to cover. So use this introductory guide to traverse the report more efficiently.

Brief Outline

The report is broken into several segments:

- “Executive Summary” on page 1 discusses the premise (and promise) of content management as a field and describes its soft and hard benefits, as well as our criteria for analyzing vendors and a glance at each one.
- “Part 2 – Content, Experience, Management, and Systems” on page 10 looks at key concepts in content, management, technology, and the business case for implementing a CMS.
- “Part 3 – CMS Dissected” on page 19 explores the web content management lifecycle, defining and demonstrating key features you’re likely to find in the technology. Then it explains our methodology for evaluating the vendors themselves. Finally, it identifies 11 universal CMS scenarios to help you “place” your own needs within the market.
- “Part 4 – CMS Product Comparisons” on page 129 begins the meat of the report. Here we categorize vendors into different tiers and then provide comparative evaluations of each product individually.
- “Part 5 – Advice, Pitfalls, and Best Practices” on page 836 offers just that, including how to select a vendor and budget properly for your project.
- “Do It Yourself?” on page 833 addresses whether or not you should implement a CMS yourself.
- If you are looking at mid-market or lower solutions, you might wish to consider Real Story Group’s list of “Regional and Niche Players” on page 834 as part of your list.
- “AJAX” on page 854, offers a short glossary, and a supplementary guide to web content governance.

Browse by Role

Depending on your role in the enterprise, you may wish to start at different places in the report and then backfill your knowledge as necessary. Consider the following shortcuts.

Project Owner

Start with the “Executive Summary” on page 1, then turn to “Part 2 – Content, Experience, Management, and Systems” on page 10 to help you distill the business case for a CMS. You’ll want to explore our discussion of universal scenarios on page 116, and perhaps the vendor comparison charts beginning on page 139. Finally, be sure to review “Part 5 – Advice, Pitfalls, and Best Practices” on page 836, before dipping into individual product chapters.

Project Manager

You might want to start with the “Executive Summary” on page 1, and then get familiar with the technology in “Part 3 – CMS Dissected” on page 19. Then use the vendor comparison charts on page 139 to consider a long list of vendors to investigate. Use the “Compare to” rows of the vendor summary charts to find other similar vendors who may not have made your long list. Finally, be sure to review “Part 5 – Advice, Pitfalls, and Best Practices” on page 836, as well as “AJAX” on page 854, especially the governance section.

Architect/Developer

We’ll assume you understand key technical concepts and want to investigate solutions straight away, if so, use the vendor comparison charts on page 139 to consider a long list of vendors to investigate. Use the summary comparison charts under each category to review basic technical compatibility. Refer back to “Part 3 – CMS Dissected” on page 19 to catch up on any key concepts that are new to you.

Marketing Manager/ Content Producer

Once you have read the “Executive Summary” on page 1, you will probably want to learn more about the technology in “Part 3 – CMS Dissected” on page 19, as well as “Part 5 – Advice, Pitfalls, and Best Practices” on page 836. Use the vendor comparison charts on page 139 to sift through your long list of vendors and compare how each offering’s functionalities stack up against each other.

Consultant / Integrator

How you use the report depends on what you need. Each section can provide useful analysis. You may want to jump to the vendor charts on page 139 to begin your investigations, particularly if you are in search of suitable software partners. Pay special attention to the “Vendor Intangibles” sections in each vendor chapter.

Part 2 – Content, Experience, Management, and Systems

This report will help you navigate the maze of web content management systems, enabling you to identify your own requirements more clearly, understand which technologies are best suited to your needs, and know the best practices to ensure that your implementation is a success.

What are content management systems, how do they support experience management, and why have they become so important? To answer these questions, this section breaks it down by looking at the concepts of Content, Experience, Management, and the Systems around them.

The Digital Content Mandate: Captivate the Visitor

Content is always king. Ask any group of content managers, and they will tell you — after investing in applications and services over the past five to ten years — they still often find themselves drowning in the mass of critical supporting content that lies beneath everything they are doing online. Pair that mass of content with the more recent focus of content management — digital customer experience management — and today’s online content and marketing managers must also manage content for different user types, in various contexts, and on various devices. What is the best way to manage content to support what customers want? How can content management be used as a tool to draw visitors into a website with content personalized specifically towards them — and keep them coming back?

In the mid-to-late ‘90s, when content management became a category of software, the focus was the content lifecycle: Create, edit, deploy, and possibly archive or delete (although arguably, not enough of it). Original website managers were happy simply to see content appear on a website, and there wasn’t much thought of the people on the other side of the screen. Indeed, content management used to be just about content, but it’s become much more than that. Content may be king, but now it’s sharing its rule with a very demanding queen: the customer or employee you’re trying to engage.

Larger and More Complex Websites

The sheer volume and increasingly rapid growth of web content creates tremendous challenges for content managers. If your own websites are growing at nearly unmanageable rates, you are not alone.

More content does not necessarily make for a better website, and some content managers find that as more material is pushed online, it actually diminishes the value of existing content; valuable material is harder to organize and is typically only found by patient visitors. A good search engine can help here, but ultimately, the ability to develop user-centered navigations and metadata schemas, as well as enforce quality control mechanisms — likely with the help of a Web Content Management System (“CMS” or “WCMS”) — becomes even more important.

One of the unanticipated side benefits of a CMS project is that it forces enterprises to reconsider the value of the content they are publishing, and in many cases, the weight of migration alone compels organizations to eliminate vast quantities of unused content from websites. Going forward, a CMS can also enable an organization to keep a better handle on unwanted content explosion.

Content and Customer Experience Management

Today many companies' digital efforts have transitioned from simple “web publishing” to more full-blown “customer experience management” (CXM), as they seek to take advantage of the Internet, as well as other channels like offline, print and mobile, to conduct core commercial operations and grow their businesses.

CXM can be defined in many ways. We use the following working definition:

A strategy and practice for managing customer experiences online (including web, email, social, and mobile) and offline (such as print, call centers, and in-person locations) to acquire, retain, and manage up customers into satisfied, loyal brand advocates and ambassadors.

The cross-organization discipline and practice of CXM defines how customers and prospects are acquired, nurtured and managed throughout their customer journeys. Many technologies in the Information Management landscape are used daily to support CXM initiatives in any given organization: marketing automation, customer relationship management, DAM, and WCXM.

Whereas in the late '90s, web managers might have overseen a single (and likely static) public website, they now must juggle the demands of an intranet, extranet, possibly multiple public corporate web properties, some of them in multiple languages, as well as the company's participation in any number of social channels. In short, corporate content now lives in a virtual ecosystem, compounded by the complexity of managing it.

More Demands

Today, more demands than ever are placed on web content management processes. Content managers must often work with highly diverse content sources, pulling text and images from adjacent DAM systems, or if that is not available — from databases and file systems around the company.

In response, some organizations devolve content maintenance to content owners. A broader group within a firm taking a bigger stake in content management can be a very good thing, and in visionary companies, this happens by direction. In most firms, however, the devolution of content maintenance occurs in an ad hoc way, and this frequently leads to quality problems if the system is not designed for distributed, collaborative content management.

Typically, issues with quality first appear at the content and presentation layers. Content becomes redundant, unapproved, unoptimized, and/or outdated. You may have editorial controls in place for offline content that is later repurposed for the web, but what about the proliferation of web-only content? Who is controlling that, and how? Navigation and usability also suffer, as page archetypes begin to break down and different layout schemes display errantly throughout the website.

Content managers also face pressure for more dynamic content — up-to-the-minute pricing, recommendations (“You looked at this, also look at that”) or personalized content. Internal stakeholders and external customers alike will frequently ask, “If it's in a database here, why can't we just push it to the web?” Unfortunately, it's rarely that easy, even though techniques for extracting and presenting database content online have become ubiquitous. However, dynamic, database-driven pages have a serious impact on download times, especially for busy sites. Competing demands for site speed and dynamic presentations, therefore, put content managers on the horns of a dilemma.

The demands of delivering content across multiple channels is another struggle for organizations. Do we use a mobile app, or a mobile site, or both? Do we need a Facebook or LinkedIn page, and how can we manage that content from within our CMS? What about workflowing tweets? Having a solid content strategy for all channels you might be engaging in — email, web, social, mobile, print, etc. — has never been more important. Your visitors have never been less patient, and are coming to your businesses from different contexts, devices, and countries, with different goals in mind. The job of today’s digital manager is to not only manage content, but also to provide content that has been analyzed, optimized, and personalized for different audiences of your business.

Publish Faster

The pressure to convert corporate information into consumable online content rapidly became a routine demand. Organizations looking to implement a CMS may be also trying to eliminate the “IT bottleneck” and have more control over content from the marketing department, for example.

Companies want to respond more quickly to market conditions and competitive pressures. This places a new, higher premium on the adaptability of website structures and presentations at exactly the same time that managers are struggling to tame publishing processes that threaten to expand beyond reasonable control.

Of course, end users want information faster, too. From manufacturing, the online world has borrowed the term, just-in-time (“JIT”) content. In a customer-driven environment, responsive companies must publish the content the customers specifically need, when they need it, in the format they need it.

Publish Cheaper, Reuse More

As their online efforts proliferate, enterprises have begun to take harder looks at the costs of their rapidly expanding infrastructure. Naturally, one of the biggest expenses is people, especially in the technology field. Savvy firms are tallying the amount of time that non-technical contributors need to expend updating websites, and are looking for ways to automate the entire process and re-use content more efficiently, where appropriate, across multiple websites and channels, such as print to web or vice versa. To be sure, cost savings can be difficult to quantify (see section on “ROI” below), but a CMS should allow you to publish greater and more complex content going forward, with the same or fewer resources than you have today. It should also enable production staff to be reoriented towards higher-value work, such as content strategy, content analysis and optimization, as well as personalization and segmentation.

Of course, the CMS costs money too. In the search for software solutions, it is not unusual to experience sticker shock at the price of CMS packages, particularly those targeted at the enterprise, where multi-server, hundred-contributor licenses can run into the millions of US dollars. Fortunately, the CMS marketplace is constantly expanding and evolving, introducing a variety of licensing models and costs, and a surfeit of vendors have emerged to address the mid-market customer, and even small businesses. (More about that in Part 4 – CMS Product Comparisons.)

Support Social Participation, Increase Customer/Employee Satisfaction

There's also the factor of content contributed by the visitors themselves, be it blog commentary, tweet in your direction (positive or negative), a Facebook "like" or "share," content edited via a wiki, or a response to one of the topics on your forum. The results of these contributions must be instantaneous, allowing customers to see the results of their actions, thereby driving their own online experiences and increasing their "satisfaction factor" with your brand or for employees, your workplace. Whatever risk to your reputation to have irrelevant content in your channels or content that is hard to navigate to or find, it might be even worse to hold comments or "wall posts" in moderation for a long time, discouraging customers from participation, diminishing their experience, and undermining their trust into your brand.

Many organizations still struggle with customer participation and social content, often lacking social media and community management strategies and proper personnel in place. Nevertheless, CMS vendors have jumped on the social software bandwagon, and in some cases, let their core WCMS product development stagnate in the process. Some vendors turned to remarket their software as "Social CMS."

So, when a vendor calls its product a "Content Management System," what does that really mean?

What Is a Content Management System (and What Is It Not)?

There are almost as many definitions of "Content Management System" as there are CMS vendors and analysts. At the end of the day, the solution you need is likely to be as unique as your business. Here is Real Story Group's definition of a web content management system:

"A system that lets you apply management principles to web content."

That might seem self-evident, but most companies today — large and small — do not manage content with the same rigor that they manage data. Under this definition, content moves to the center of the equation where it belongs. As in all management systems, people play a decisive role in what happens to it. What they do with content can be encapsulated into business rules and editorial processes. The goal of these efforts is to support specific business objectives.

In short, content management products are not in themselves content management systems. Nevertheless, technology is required to support your rules and processes. Of course,

technology costs money. Even if you just reorganize your system without applying new technology, you will expend precious resources (time, attention, and probably money, too). Therefore, a smart company will ask, "What is the business case for overhauling our content management system?" The next section addresses this question.

The Business Case: How to Justify a WCM Investment

At some point, someone will ask you to justify your Web Content Management System (WCMS) project. Fortunately, there is a business case to be made for a WCMS on both the cost and revenue sides of the ledger.

If no one is requesting a business rationale and the project is proceeding regardless, reconsider your bearings. All major technology efforts beget tough decisions and difficult

trade-offs, so it is best to have a business plan in place as a touchstone to keep the team focused — especially if your CFO will be writing big checks to outside vendors.

Like any investment, a WCMS project can be justified in terms of “hard” benefits, which are typically more easily quantified, and “soft” benefits, which may be more qualitative, but they are certainly no less important. Additionally, like any IT project, nearly all business benefits flow from concurrent organizational change, so for each advantage, there is an important caveat or two.

Hard Benefits

Let’s address the financial side of the equation first. A new CMS can enable you to:

- **Increase transactions**

A CMS can help you publish better and more information, providing better context around products and services for a more differentiated, solutions-oriented proposition. By publishing richer, more accurate, and higher-quality information online, you may be able to increase visitor interaction, whether it is buying a product or inquiring about a service you offer, or transitioning costly offline HR transactions to an intranet.

Caveat: More content won’t help unless you take the time to organize, analyze, optimize, and personalize it better.

- **Accelerate your time to market**

Printing and distributing marketing collateral, sales catalogs, and customer support documentation takes time. Effective content management, coupled with content delivery, should enable you to reduce the travel time from your content owners to content consumers among your prospects and customers. This speeds the return on your investment and improves cash flow — music to any CFO’s ears!

Caveat: A Web CMS will not provide a suitable single-source repository for all of your digital content.

- **Achieve process efficiencies**

An effective content management system should enable you to spend less time on web production and updates, with less effort propagating changes throughout multiple versions of the same content, and fewer cycles editing different iterations of the same content. The right CMS may also help you achieve better team collaboration and faster content approval processes.

Caveat: A non-intuitive system will erode any efficiency gains; heavy-handed workflows can reintroduce bottlenecks.

- **Increase customer acquisition and retention rates**

By producing and tagging more relevant content in the CMS, you can provide your customers with better information, which in turn will help them make a buying or other decisions. Doing this well better positions you to not only acquire new customers, but to turn your existing customer base into loyal advocates of your brand.

Caveat: A Web CMS alone will not optimize and personalize your content to keep existing customers happy, or to reach out to new audiences. This work requires experienced editors and marketers, under a sound strategy.

- **Increase customer satisfaction and support**

There are multiple ways you can support your customers from a content perspective. Forums, blogs, wikis is one way. Documentation is another. Social media channels is

yet another arena where customers tend to reach out to companies for support with issues or questions, praise or complaints. A CMS can serve as a central repository of content and a single vehicle for content delivery, as well as a place of a unified, consistent, and approved corporate messaging.

Caveat: Some types of highly-structured documentation may be better managed in a Component Content Management purpose-built for XML-based component reuse.

Hard figures on costs savings are difficult to come by, and results will vary. Many CMS vendors offer “ROI calculators.” These can be helpful to orient your thinking, but a savvy manager will take their assumed labor savings with a grain of salt. ROI calculators also chronically underestimate the additional effort involved in implementing and then managing the CMS, itself.

Unlike other content management efforts — such as imaging projects — a CMS system typically won’t reduce head count (although it can help you refocus existing staff). A CMS yields real cash savings, most commonly where an enterprise is presently outsourcing web production.

In general, the promise of financial benefits of a CMS must be tempered by the ongoing requirement to pay for maintaining and improving the CMS itself, as well as the potential for cost overruns, if the solution is mismatched to your needs (a woefully common experience).

Soft Benefits

Some of the intangible benefits of implementing a CMS also count among the more powerful rationales for making the investment. A CMS enables you to:

- **Put businesspeople in control of your digital communications**
Web publishing efforts are too often marginalized to distinct cubbyholes within companies. Greater automation and non-technical interfaces give marketing and businesspeople more control of what is published, when, and how. This invests them more in your company’s online success, and supports better content quality.
Caveat: You may need to persuade reluctant content owners to participate, now that web publishing is no longer “someone else’s problem.” You may also need to persuade your IT department to step back and allow editorial contributors to take control of what used to be code.
- **Maintain user experience and brand consistency**
Experienced marketers know that it takes real work to maintain the clarity of a user’s experience and the consistency of a company’s brand. By separating content from presentation, a CMS can enforce visual consistency through a series of controlled templates. Light Digital Asset Management capabilities within a CMS are now commonplace, so that worldwide companies can centralize brand standards and guidelines to a single source. Customers are more demanding than ever, and expect targeted content — relevant to their personal preferences and previous interactions — and many CMS tools have the ability to deliver personalized content to specific segments of your audience.
Caveat: Your brand is much bigger than look and feel, and a CMS cannot enforce the quality of your content.
- **Improve your agility**
As markets change and customer bases shift, your communications efforts need to

respond in kind. If your CMS places structured content in an XML repository or database, for example, you will better position your enterprise to participate in various marketplaces, future syndication projects, and other collaborative ventures.

Caveat: It is difficult to truly separate content from context for maximum content reuse.

- **Improve security**

Implemented properly, tighter control of your web publishing efforts can improve the security of both your systems and your content. For starters, it can help you control who gets to work on what content and layouts. Perhaps more importantly, you can reduce the number of staffers touching production environments by authorizing only the content management system to promote content to live servers.

Caveat: Like all IT systems, a WCMS brings its own vulnerabilities.

- **Improve record keeping**

In an era of heightened awareness and new regulations concerning the maintenance of proper electronic records, a content management system can provide an essential audit trail of what was published when, where, by whom, and on what authority. A CMS can also reduce the effort associated with recreating an older version of the site for regulatory or legal reasons.

Caveats: You need to make sure that your CMS actually can audit all of this activity; many CMS tools remain incapable of key records management concepts like retention schedules.

- **Innovate and improve your competitive stance**

By freeing you up to focus on content and experience quality, you can better out-compete online. The right CMS should also facilitate experimentation and metrics, so that you can innovate faster.

- **Maximize skills and talents through specialization**

A good CMS enables specialization, wherein designers concentrate on the user experience; engineers zero in on business logic and code; and content owners focus on the quality and relevance of the information they create and maintain. Along the way, the company obtains improved accountability and employees provide more value.

Caveats: “Generalist” webmasters may resist this change; managing the CMS, itself, requires specialized skills.

Which Rationale?

There are different ways to build a business case for implementing any technology solution. One approach looks at Return on Investment (“ROI”) or its cousin, Internal Rate of Return (“IRR”). This approach entails demonstrating the financial return on your investment, and is a very powerful justification — if you can credibly predict attractive financial returns.

Another approach classifies a major expense as a necessary precondition for doing business — such as investing in an upgraded phone system — in circumstances where there is no immediately definable ROI. This is known as Cost of Doing Business (“CDB”) analysis. A common and quite successful business rationale for a WCM is, “*We just couldn't go on without one anymore.*”

Both approaches offer valid analytical models for constructing a business case. The paucity of ROI justifications for web content management has led some commentators to downplay Web CMS projects in the wake of the dot-com bust. However, the enduring demand for WCM

solutions suggests that enterprises view them as a necessary cost of communicating effectively to customers, partners, and employees alike.

	ROI / IRR	CDB
Complete Name	Return on Investment / Internal Rate of Return	Cost of Doing Business
Definition	Calculate the value of an investment by dividing incremental revenue + cost savings by the amount invested	For any company, certain investments must be made as a precondition for successfully conducting business
Classic Example	Rolling out a traditional direct marketing campaign. Marginal sales increases can be measured against incremental costs.	Firewalls. You must connect to public networks to do business, but also secure your information systems from outside intrusion.
When to use?	Use when there are clear cause-and-effect outcomes and incremental revenues or cost savings can be readily quantified. For IRR, look at the opportunity costs of not implementing a CMS.	Use when key arguments cannot be quantified as readily. CMS is a precondition for other corporate revenue and cost-cutting initiatives, rather than vice versa.
Possible CMS Benefits	Greater sales, broader offerings, reduced customer service costs, reduced content production costs, reduced legal liability, faster time to market and adaptation	Wider exposure, clearer brand identity, more fulfilled and specialized staff, greater customer loyalty, broader future business opportunities, less risk of public failure
Decision-makers likely to concur	Finance-oriented, data-driven	Strategic, visionary, intuitive

When All Else Fails

What if you made your best case, and you still didn't come up with funding? Try pulling one of these three arrows from your quiver.

- First (as any software vendor will remind you), in many situations, content management systems can become fundamentally an *infrastructure* play, so the financial resources applied towards building a new system can often be capitalized on the books. This is especially the case if you are implementing a new system for legal or regulatory reasons.
- CFOs are more leery about this in the environment of investor and regulatory scrutiny, but it is still possible to find legitimate ways to amortize expenses. One approach is to consider using a hosted service (described later in this report), where the costs (and actual cash outlays) are spread out over the life of the application, rather than paid entirely up front.

- If fear of litigation or regulatory non-compliance is a driving factor in the choice to move forward, then you can justify the amortization expenses as akin to an insurance premium.

On the other hand, if you cannot secure funding and executive support, reconsider your bearings. Without senior leadership, a disruptive technology project (that certainly includes a CMS procurement, implementation, and adoption) is unlikely to succeed. Focus instead on improving your content and processes with the resources and tools already at your disposal.

Part 3 – CMS Dissected

This report focuses on Web Content & Experience Management (WCM — also referred to as “Web CMS,” “CMS,” or “Web Experience Management System”).

Content Management Basics

Here we will try to break down some of the basic business functions a bit more to begin to isolate the core features of web content management versus related disciplines, such as document management and digital asset management.

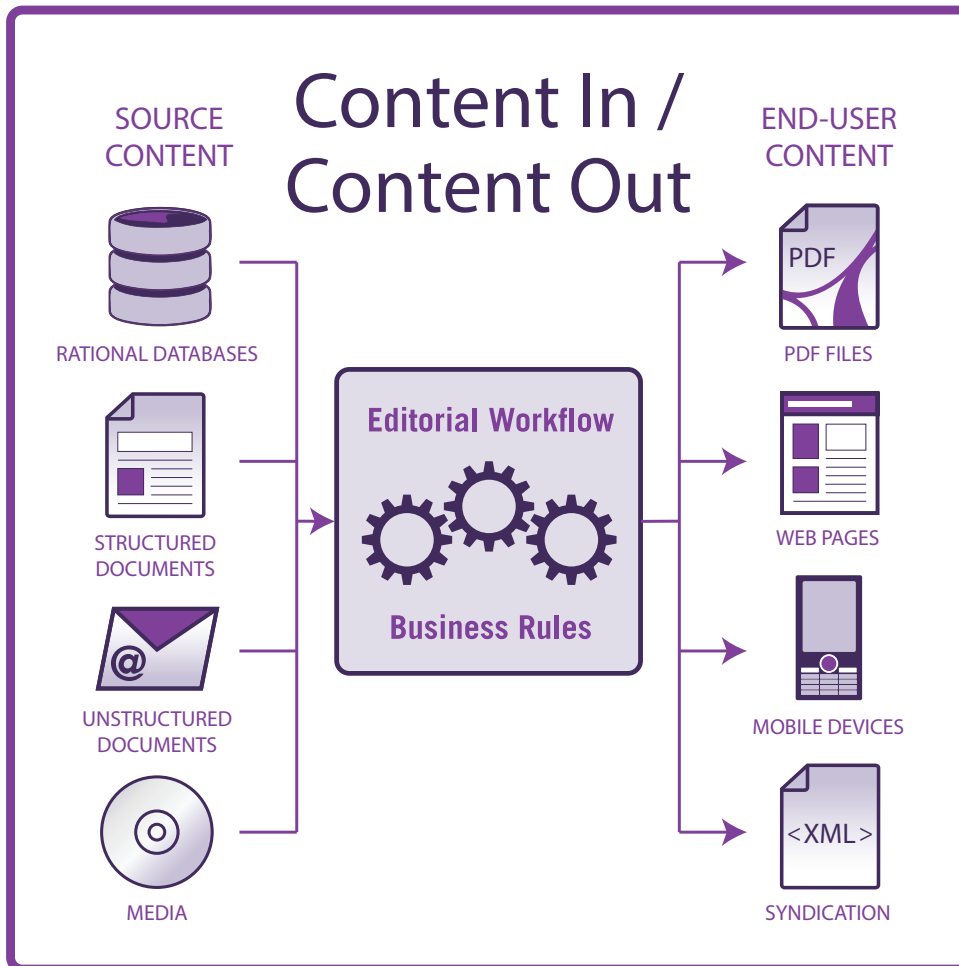


Figure 1. Content in / Content out.

At a very basic level, all content management systems do the same thing: Take in content, add value to it by applying approval and other business processes, and then output it in some format. The simple input/output chart (Figure 1) can be applied to nearly any class of content technology.

Within a content management system, regardless of content type, several standard features are typically available to support these business processes. Whether the content in question is text,

images, binary documents, XML nodes, multimedia files, forms, or something else, these core capabilities are essential in any content management system:

- Contributor and managerial **rights and privileges** must be managed, usually according to preset roles. This promotes security and ensures that participating staff people are only undertaking suitable and appropriate tasks.
- Content must be **authored or ingested** into the system, and sometimes transformed into a consumable format. This enables corporate information to be actively managed.
- **Repositories** must be managed, through versioning and version control. This ensures the integrity and authority of the core content.
- Content must be **tagged with metadata**. This enables content to be subsequently retrieved more easily and reused more widely, with minimal human intervention.
- **Workflow** mechanisms must be in place. This helps assure consistency, quality, auditability, and reliability of content and business processes, alike.
- Content must be **localized** for multiethnic or multilingual audiences, as well as authors. This enables enterprises to extend their content management efforts across national boundaries.

However, the function points of web content management (as opposed to document management or digital asset management, for example) still represent a distinct solution set, with its own unique business and technical drivers. Before we explore Web Content and Experience Management in more detail, let's look at some adjacent technologies.

Portals

Portals are designed to aggregate information and services. HR handbooks may reside in word-processing files that could easily be indexed and shared within a CMS. However, what about corporate pronouncements or other content residing in email archives (where much knowledge lives in the contemporary corporation), or sales data buried within your ERP system? To integrate that all together — at least at the interface or presentation layer — you may need a portal.



Portals

At their best, enterprise portals serve as the end-user prism into complex corporate content. Then an intersection with WCM becomes readily apparent. Content still remains at the heart of any portal, and therefore the management of that content, including versioning, workflow, and presentation control — all typical WCM features — is required. Without a CMS, an enterprise portal is “read-only.”

Digital Asset Management (DAM)

The business case for DAM traditionally argued that companies whose lifeblood revolves around their multimedia digital assets — such as entertainment and media companies — should actively organize and repurpose those assets to streamline costs and enhance revenues.

These systems are especially suited to managing multimedia content, such as images, photographs, or video. Unlike other products, DAM systems tend to offer hooks into specialized creative, editing, and design



Digital Asset Management

applications, and cull specialized metadata from multimedia assets. If multi-page product marketing brochures that were designed in Adobe InDesign are your main web content, you may want a DAM instead of a CMS (or sometimes in addition to the CMS). If multimedia content serves as your company's products itself — rather than supporting other products — then you almost surely want a DAM system.

An important choice for you to make is whether you need to develop a management system for your media assets at large — regardless if web delivery is the principal outcome and channel. If the answer is “yes,” you may want to look into a DAM package. If the answer is “no” or “maybe,” the asset management features found in a web content management product would probably suffice. Many WCXM products provide fairly light asset management capabilities, like in-browser photo gallery creation, image cropping and scaling, and the ability to incorporate videos into your content.

Digital Marketing

Digital marketing tools span a wide spectrum of capabilities, from email campaign creation and management, to social media marketing and analytics, to landing page management. While some CMS vendors pack their products with digital marketing features, it is important to know where to draw the line and turn to specialized tools. In general, WCM technologies can support inbound marketing, and digital marketing platforms emphasize outbound marketing.

Digital marketing tools often have some notion of a marketing data warehouse. WCM tools typically are not ideal for storing customer and prospect data. However, both systems typically need to connect to your CRM system for extant customer data.

Web Content & Experience Management (WCXM)

Major web content management packages typically offer core information-management services, such as versioning and workflow, discussed above. Additionally, they add special capabilities to the mix, including, potentially:

- **Specialized Authoring and Transformation Tools:** To enable business users to input content into the system and have it normalized to HTML or XML
- **Aggregation and Component Management:** To combine and publish discrete chunks of content that may originate from a variety of sources
- **Templating:** To ensure consistent, predictable renderings for the web and, potentially, email and mobile channels
- **Deployment Path:** To publish to standard deployment chains of environments (development, stage/testing, and production)
- **Page Assembly and Delivery:** For dynamic production and submission of content to end users (content consumers)
- **Segmentation and Personalization:** To deliver targeted sets of personalized content to specific customer segments
- **Caching and Replication:** To ensure high performance in public environments characterized by spikes in demand

- **Engagement and Social Media:** To provide basic interactivity to a website and to engage your visitors. Includes functionality such as wikis, blogging, commenting, microblogging, and social media integrations (e.g., Facebook, Pinterest, or Twitter)
- **Optimization and Testing:** To analyze, test (A/B and MVT) and measure published content in order to optimize it further
- **Digital Marketing and e-Commerce:** To create and manage promotional campaigns and integrate with e-commerce, CRM and other related systems such as Salesforce for various types of automation
- **Syndication:** To add value to content through advanced Internet-based distribution
- **Producing Mobile, Tablet, and Other Experiences:** To create and manage content and experiences specific to various mobile devices and tablets

Standard CMS features are detailed in the next section.

How CMS Tools Work

We analyze Web CMS tools across three different sets of services:

1. Technology, including development and administrative services, as well as performance and scalability
2. Content, including authoring, UI usability, taxonomy management, workflow, and globalization
3. Experience, including site & campaign analytics, mobile, and social media integration, among others

Some CMS tools offer only a subset of these services, and in these cases, it might make sense to integrate your CMS with separate tools. Nevertheless, this is the holistic perspective from which we approach web content and experience management, incorporating both content and experience management capabilities.

By understanding your specific needs and opportunities in each dimension, you can begin to construct the outlines of a new CMS. You will also have a roadmap for evaluating the suitability of various package offerings.

In the pages to follow, we will identify the key system attributes for all three dimensions that you should consider when developing your requirements.

Dimensions of Web Content & Experience Management		
Technology Services	Content Services	Experience Services
<p>Technical Administration & Security</p> <ul style="list-style-type: none"> • Threat Prevalence • Authentication & Authorization • System Reporting • Multisite Management • Cloud Services <p>Development</p> <ul style="list-style-type: none"> • Configuration & Customization • Integration & Extension • Content Modeling • Templating <p>Performance</p> <ul style="list-style-type: none"> • Back-end Performance • Site Caching & Delivery 	<p>Contributor Experience</p> <ul style="list-style-type: none"> • Overall Usability • UI Accessibility <p>Contributing Content</p> <ul style="list-style-type: none"> • Authoring & Transformation • Tagging & Taxonomy • Content Reuse • Media & Document Management • Repository Services <p>Content Lifecycle</p> <ul style="list-style-type: none"> • Workflow • Globalization • Archiving & Compliance 	<p>Publishing</p> <ul style="list-style-type: none"> • Standards Adherence • Multichannel • Mobile <p>Digital Marketing</p> <ul style="list-style-type: none"> • Site & Campaign Analytics • Testing & Optimization • Personalization & Segmentation • Social Media Integration • Promotional Campaigns • Community & UCG <p>Workplace</p> <ul style="list-style-type: none"> • Collaboration & Networking • Dashboard <p>Ancillary</p> <ul style="list-style-type: none"> • Site Search • Online Forms • Module Ecosystem

If you already have a good understanding of the concepts in the chart above, you can consider skipping this section and head straight to the product surveys. *Note, however, that the product section of the report makes heavy use of the concepts and terminology discussed below.*

Technology Services

This section deals with the underlying plumbing (e.g., access control), as well as the overlying layers of the tools (e.g., usability and internationalization). It is designed to be read by an architect or developer, although business managers should take note of usability issues and graphic designers will want to investigate templating.

Standards and Guidelines: Aligning with the Rest of the Industry

Consultants and vendors love to talk about standards and adherence to various web guidelines; it makes our industry sound more mature than it really is. To be sure, standards are important, however, there are fewer of them than you would think, and they are less standardized than you would like.

First, let's get the terminology straight and identify some important distinctions:

- **Standards:** These are a set of specifications, often with a reference implementation and usually supported by multiple vendors.
- **Guidelines:** These are recommendations or best practices. An example of this would be the Web Content Accessibility Guidelines (WCAG), which define good practices for improving accessibility of content.

Standards are quite helpful and have many advantages. Some of these benefits are:

- Integration and interoperability
- Skill rationalization
- Portability
- Simpler and more predictable development lifecycle

A by-product of all this is that development and maintenance becomes simpler and more predictable, in turn decreasing costs. In fact, if products comply with standards, you can actually do a lot of development on open source products and then deploy on commercial products. This means savings in licensing costs as well. Now developers can focus on actual business, instead of spending time on commodity tasks like browser testing.

However, as with everything else, you need to be ready to face trade-offs as well. Standards can be immature and shifting — and various factors (mostly business rather than technical) can inhibit widespread acceptance. Reduced acceptance naturally reduces their value.

Treat standards as a means to an end and not an end in themselves. Look at the standards holistically, in order to ascertain how much benefit you're likely to get and at what expense. Many people believe that a standards-based approach is always a better approach, but by asking the right questions, you can better assess which ones will provide the most “bang for the buck.”

The most important thing to keep in mind is that, whatever boxes vendors check, standards support tends to be a relative attribute rather than an absolute. There are different ways to comply with J2EE or support XML. The key question here, as elsewhere, is *how* a particular tool supports a particular standard.

In the rest of this section, we'll outline some standards that are worth exploring before you begin (or re-embark) on a Web CMS journey.

JSR 170/283 and JCR 1.0/2.0

JSR 170, which was succeeded by the newer JSR 283 in September 2009, — also known as “Java Content Repository” or JCR — is an infrastructure specification developed under the Java Community Process (JCP) program.

The idea is to provide greater code portability above the repository layer. Today almost every content management application ships with its own (frequently proprietary) “content repository.” Repository services — for example versioning — are implemented differently by nearly every vendor.

JSR 170/283 promises the Java world, and possibly beyond, a unified API that allows accessing any compliant repository in a vendor- or implementation-neutral fashion, leading to

the kind of clean separation of concerns that characterizes modern IT architectures. Some people call JSR 170/283 the “JDBC of Content Repositories.” There are long-term benefits here, potentially in connection with swapping out applications and suppliers in the persistence tier without unduly upsetting the content applications you’ve built on top.

The JSR 170/283 API defines how an application and a content repository interact with respect to a number of content services. For example, the versioning facilities of a content repository are clearly defined, so an application knows how to browse the version history, check-in and checkout content items, or update and merge content in a standard fashion.

There are different levels of JCR compliance, and not all vendors who say they support the standard fully support it at all levels. In particular, you may want to know if they support the application level (that is, their repository can be shared with other applications through a consistent API), or repository level (that is, their application can use other JCR-compliant repositories). It depends, as you can see, on just what is getting swapped out.

You may also want to check which version of the specification a vendor supports, as many have decided not to upgrade their reference implementations to JCR 2.0. The underlying architecture of JSR 283 is very similar to JSR 170, yet some new features, fine-tuning, and deprecations have been introduced after four years in the making.

JSR 168/286

This is the major Java portlet standard, and therefore more germane to portals rather than Web CMS tools, but it is still worth mentioning.

A JSR 168/286 portlet should, in theory, run inside any JSR 168/286-compliant portlet container. That means that when a CMS vendor provides a JSR 168/286 portlet for accessing its services or repositories, you should have some confidence that it will work in almost any major Java portal platform. You can actually write once and re-use existing infrastructure and applications, even in the event of migration to a different technology.

As a practical matter, portal vendors implement JSR 168/286 differently, and frequently have “extended” the standard with their own individual hooks – especially, due to the limitations of the older 168 version. Therefore, a portlet may behave differently from one portlet container to the next, and you’ll want to test any portlet carefully before marrying a prospective Web CMS tool to your portal (or vice versa).

Many standards constantly continue to evolve on their road to maturity. Therefore, there is also the risk that comes with using an immature and incomplete standard. The Java Portlet API, defined by JSR 168, is a good example of a standard that lacked important features like inter-portlet communication. This had to be fixed using proprietary extensions (making portlets *not* portable), or wait until JSR 286 came out. However, some vendors never or only partially transitioned to the newest version of the portlet standard. Even when a standard is quite mature, it is not always clear what is mandatory and what is optional.

Some CMS vendors that also deliver content are beginning to support JSR 168/286 from the delivery side. That is, they can consume as well as produce portlets, potentially obviating the need to invest in separate portal software.

RSS and ATOM

RSS and ATOM are alternative syndication formats. Both are lightweight and designed for the one-way distribution of titles, links, and optionally, descriptions or full text.

Although designed for feeding news headlines (see sample below), RSS is well suited to any circumstance where you require a lightweight format and want to drive visitors back to your website. The feed contains a series of news items, each with an article headline, short blurb, and hyperlink back to the complete document that resides at the syndicator's website.

Sample XML Format for RSS Syndication

```
<?xml version="1.0">
<news-feed>

<channel>Content Management</channel>
<url>http://www.realstorygroup.com/news.xml</url>
<date>2014-MAR-04</date>

<news-item>
  <title>Vendors Agree on new Standard... </title>
  <desc>Several major enterprise software...</desc>
  <url>http://www.realstorygroup.com/News/... </url>
</news-item>

.....
</news-feed>
```

Here, as elsewhere, you'll want to clarify what you want in terms of RSS support. If you plan to syndicate using RSS, wouldn't it be nice if the CMS package offered a mapping interface or some other tools to automate the set up for you? Sure, any CMS package can generate an RSS file (you just create a template), but how much work do you have to do to make that happen?

Similarly, almost any CMS can be made to ingest and transform RSS feeds, but only a handful come out of the box with RSS-handling tools for non-technical managers to use to control and deploy incoming feeds.

Web Services vs. REST

Web Services refers to a related set of protocols and technologies that are designed to enable applications to expose discrete features to each other (according to certain rules) over the web. Instead of hypertext, think of *hyperservices*.

Web Services are built on three interrelated standards:

- **SOAP:** The transport protocol that enables disparate applications to plug into each other seamlessly as services
- **WSDL:** The language for describing those services
- **UDDI:** A directory protocol for listing those services

All of these standards use XML.

“REST” stands for Representational State Transfer, and defines key principles for using HTTP and URI standards via standard methods. REST is another approach to providing and consuming discrete services using Internet protocols. Specifically, you serve or invoke any function through a URL that can be called. A WCM platform with a REST API might let you invoke any item in a repository through a URL/URI, and access alternate versions or kick off a workflow by adding additional parameters to the location.

Architects increasingly appear to favor REST-based approaches, largely due to their simplicity and comparative lack of overhead. Some CMS vendors, however, still equate services-oriented approaches with Web Services, and have been somewhat slower to join the REST bandwagon.

In the meantime, many vendors today can produce Web Services, but their products (with rare exceptions) cannot consume them. This means you can get more mileage out of their tools (perhaps at additional expense to the extent you need more CPU power), but rarely will you be able to swap out subsystems like workflow with service components from other vendors.

XHTML

XHTML is notionally HTML that conforms to XML syntax. In fact, it’s actually quite a bit more than that; W3C maintains the various XHTML specifications (there are several versions, and flavors within versions), which are lengthy and highly technical. The principal advantage of XHTML is that, since it is XML, it can be validated, parsed, and processed using familiar XML tools and APIs. The down side is that in order for XHTML to be valid, it has to adhere to a rather strict syntax. For example, things like upper-case element names, unquoted attribute values, and unclosed tags — as well as scores of other syntactical indiscretions that are all too common in the HTML world — simply are not allowed in XHTML.

When web designers and production specialists talk about “standards-based layout,” they usually mean using XHTML, plus Cascading Style Sheets (CSS) to control presentation and behavior. At the very least, they mean using CSS to control the positioning and padding of elements rather than 1-pixel wide GIFs or tables within tables. It increasingly also means supplying style information in external CSS files rather than inline, in the HTML itself. Full XHTML compliance demands much more than this, but you get the idea.

Some CMS packages generate non-standard markup, and most will allow authors and other end users to override generated markup with hand edits. The widely varying quality of HTML produced by WCM systems is partly due to the variety of rich text editors the various offerings embed and the degree to which they can or can’t be locked down to prevent users from inserting their own non-standard markup. However, in other cases, CMS vendors will auto-generate perfectly legal HTML (or even XHTML) containing table-based layout, or other transgressions that may simply be “worst practices.” Be on guard for this, especially when a vendor promotes drag-and-drop layout or template development. Do a source dump of a page now and then to see what the markup looks like.

It’s not always easy to get into the guts of the system and change the behaviors that lead to inferior markup. (You can generally alter the system’s behavior here, but it can be quite a

hassle in some cases.) As always, test with your own templates and content before signing any agreements.

Recently, some vendors have begun to embed their own XHTML validators, or link to third-party validation services. Even so, you should test your pages in recent versions of all the major browsers, on all applicable operating systems. Internet Explorer is famous for not liking most varieties of XHTML.

HTML5

HTML5 is the latest reincarnation of the HTML standard. The newest revision came in response to the evolution of the web itself, with increasing use of multimedia, video, semantics, and more complex web applications. In addition to cross-browser interoperability, HTML5 also explicitly addresses the challenge of delivering content to various mobile devices and tablets.

The specification is a work in progress under W3C auspices. Its current scope limits the specification to providing a semantic-level markup language and semantic-level scripting APIs for authoring accessible web pages¹.

Many presentational features of HTML have been changed in HTML5, to name just a few:

- Specific audio and video tags (<video>, <audio>)
- New semantic web elements with <section>, <article>, <header>, and <nav> tags
- New attributes for web forms with <input> tag
- Drag-and-drop and <canvas>

A slew of new elements, attributes, syntax changes, and character encoding changes have been introduced by HTML5. In addition, there are new APIs for creating web applications.

One of the most important things to note about HTML5 is that in its current state, you will see that not all web browsers support all aspects of the standard. However, some WCM vendors are starting to bundle HTML5-capable rich text editors.

XML

XML has a natural place in web content management; it was originally designed to describe relationships within structured content. The case for XML support is manifold:

- **XML is based on common syntax, but no preconceived semantics.** That means everyone who uses it needs to follow basic rules (that makes it portable), but you don't have to bend your business to a predefined data model. Your content will probably have its own unique structure and XML will extend with you.
- **XML facilitates data interchange.** Disparate systems and enterprises can share content via XML without having to expose internal data models or invest in complex integrations. Therefore, XML is most useful for "data in motion." Similarly, this means that if your content is going to be delivered in only one format (e.g., web), there is no need to invest in XML — it will add little or no value.

1. Source: <http://dev.w3.org/html5/spec/Overview.html>

- **The “eXtensible” approach typically enables more granular control and adaptation.** The holy grail of content management is separating content from site map (“where it lives”), as well as separating content from layout (“what it looks like”). This enables you to repurpose and redeploy the same content to multiple locations, devices, and skins. XML does precisely this: It tells you what the content is, not where it resides or how it appears. Databases can accomplish this too, but XML is better suited to hierarchical content structures that you typically find in text documents (as opposed to relational structures that typify catalogs).

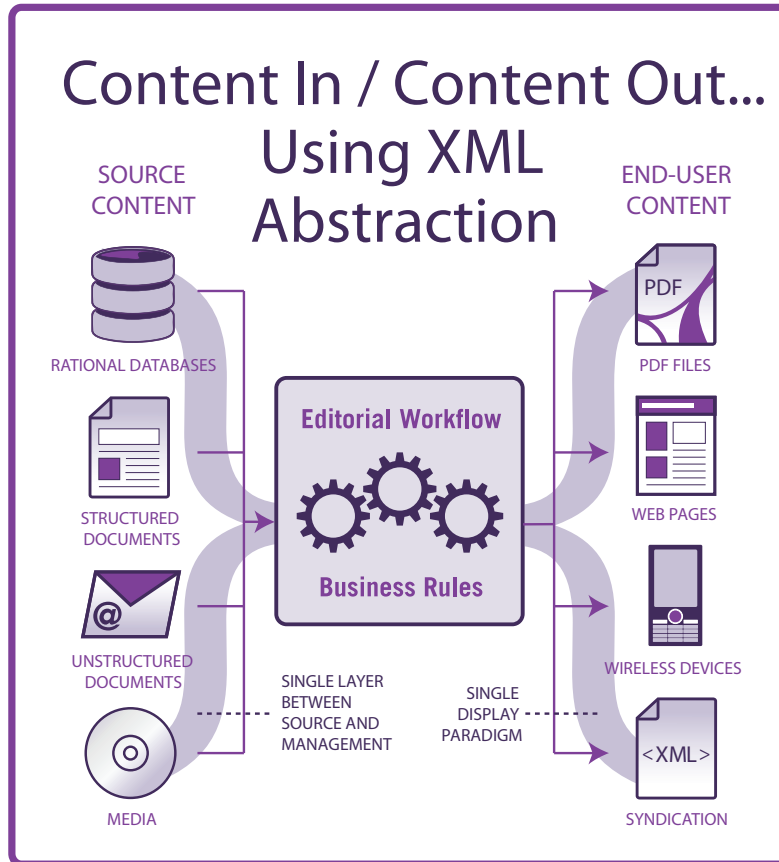


Figure 2. Content in /out, using XML abstraction.

- **XML has become the “Lingua Franca” for aggregating disparate content elements.** XML makes it substantially easier for web publishers to assemble atomic bits of content in an organized way within one site, or indeed, on a single page.

In short, XML can add value across the WCM lifecycle. So not surprisingly, nearly all content management vendors will assert their support for XML. However, Web CMS products often differ substantially in their XML capabilities. Some key issues include:

- **XML ingestion:** Can the package natively import an XML file or message stream into your system? What tools are provided to make this easier for you to map against your own content store? Note that if you intend to use syndicated content, this could be very important to you.

- **XML storage:** Does the product store XML content? How? Note that some vendors that leverage XML substantially don't actually store content as XML.
- **Transformation capabilities and tools:** If XML is not meant to be consumed, then you need a way to transform it into something useful. Ideally, the actual CMS package provides transformation capabilities natively so you don't need to go outside the content management system to render your XML into something presentable (like HTML). Conversion of desktop documents to XML is particularly valuable. Of course, all the caveats about Word from this report's section on Authoring and Transformation still apply.
- **Intelligent management of XML documents as XML documents:** Many packages can put text files (including XML) into their workflow, version them, and perhaps export them. Fine. However, you'll want to ask whether the system can interrogate those documents, understand their inherent structure, and do useful things with them based on their semantics (such as recombine elements from different XML documents into a new document). Does the platform give you programmatic access to XML nodes? Will it allow you to package up documents at different levels within your tree? Many CMS packages can't do either one.

WCAG and Section 508

WCAG is shorthand for the W3C Web Content Accessibility Guidelines. As the name suggests, these are not really standards, per se, but rather a set of guidelines to follow to make your pages and applications more accessible. We think of them collectively as a kind of standard, and so should you. Section 508 is a US federal government regulation stipulating that federal agencies must give disabled employees and members of the public access to information that is comparable to the access available to others. It has analogues in different countries around the world.

WCAG is quite involved. There are four guiding principles in WCAG 2.0:¹

- Perceivable
- Operable
- Understandable
- Robust

Each of the above includes a number of very detailed guidelines on how to treat text alternatives, time-based media, layouts, use of color, visual presentation, keyboard accessibility, images, navigation, and similar issues properly.

In order for a web page to conform to WCAG 2.0, it must meet specific conformance requirements. One of the three levels must be met in full:

- Level A (the minimum level)
- Level AA
- Level AAA

Clearly, a CMS can help here, and some vendors will boast that their tools are WCAG compliant (by which they usually mean the pages they publish). However, true accessibility is really up to you. (See section on “UI Accessibility” on page 62 for more on *interface* accessibility.)

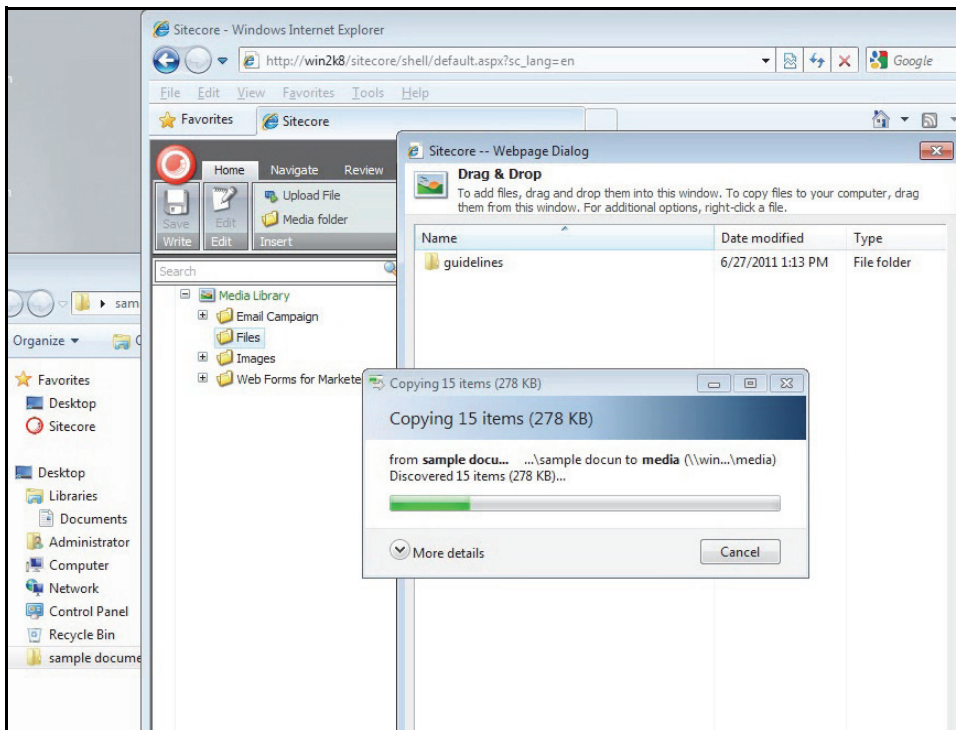


Figure 3. Checking in an image from Photoshop using WebDAV.

1. Source: <http://www.w3.org/TR/WCAG20/>

WebDAV

WebDAV stands for “Web-based Distributed Authoring and Versioning.” It is a set of extensions to the HTTP protocol that allows users to edit and manage files on remote web servers collaboratively, without overwriting others’ work. WebDAV-compliant CMS products can enable contributors using similarly compliant desktop tools to check files in and out of server repositories directly from the desktop, avoiding proprietary plugins and messy (and likely insecure and unsynchronized) file transfers. Most HTML editors, for example, are WebDAV compliant. Alternatively, you can use Windows Explore to drag files into a WebDAV repository.

In addition, WebDAV enables an associated XML-based metadata payload that can ensure that certain key attributes (like author, expiration, etc.) will be maintained and updated along the way.

WebDAV compliance can vary from implementation to implementation, so test before you plunge, but the protocol can be extremely helpful, especially for image management (where the atomic unit for web publishing purposes tends to be a discrete file). In Figure 3, an author is checking a document into a Sitecore repository.

Content Management Interoperability Services (CMIS)

Content Management Interoperability Services (CMIS) is a standard that was ratified by OASIS (Organization for the Advancement of Structured Information Standards) in May 2009. The CMIS specification is designed to enable interoperability of content management applications at the repository services layer. Adobe, EMC, IBM, Microsoft, OpenText, Oracle, and SAP are some of the original backers of CMIS.

CMIS uses wire protocols rather than direct programmatic calls (like JCR) or special remoting protocols of the RMI or CORBA type. Client applications talk to CMIS servers via Web Services (e.g. SOAP) or AtomPub protocol bindings. Both a SOAP binding and a REST binding should be present in a CMIS-compliant repository.

CMIS’ purpose is to enable content management applications to talk to any compliant repository without knowing any vendor-specific details about that repository. In version 1 of the specification, CMIS can make a variety of simple CRUD operations (create, read, update, delete) on folders, documents, and repository relationships — as well as filing, search, versioning and navigation — a good deal easier for client applications that need to operate against a compliant repository. Nevertheless, the devil is in the details, and many problems remain to be ironed out in the upcoming updates to the standard.

You should note a couple of things about CMIS. One is that the backers of it are primarily ECM vendors; therefore, currently, this is primarily a Document Management and not a web content management standard. One area in which this shows up is its domain model for data, which is distinctly document-based and assumes a relational model rather than the XML-friendly hierarchical model upon which JCR (for example) is based.

Many working integrations that produced CMIS-compliant repositories and CMIS clients (IBM CMIS Firefox connector, xCMIS, Flex+AIR-based CMIS Spaces, etc.) already exist.

There are quite a few PHP, Python, .NET (DotCMIS, NCMIS, CMIS4SharePoint) and Java (e.g., Apache Chemistry and OpenCMIS) reference implementations out there as well.

Apache Chemistry was promoted by the Apache Software Foundation to a top-level project. Chemistry's open API to CMIS repositories and a set of libraries (Java, PHP, Python and .NET) is available under the Apache Software License version 2.0.

CMIS is worth keeping an eye on, especially from the perspective of how major WCM players will implement it, but don't rush to include it on your next RFP or expect all the kinks to be worked out before more mature versions of the standard see the light of day. Until then, if you have non-trivial interoperability needs, consider JCR connectors or bite the bullet and plan to use other integration methods to stovepipe together a solution.

Web Experience Management Interoperability (WEMI)

WEMI is an OASIS technical spec in very early stages of development. The spec was started in late 2011 by a group of WCXM vendors in response to market needs for more standardization in experience management technologies, as well as to address shortcomings of CMIS for certain web content and experience management scenarios.

The goal of WEMI is to define “a simple domain model for delivering aggregated content into a total web experience,”¹ and abstract a feature set that is common to WCXM systems. The first deliverable is set to target use cases around mashup content, content indexing and metadata, and content migration.

At this point, you should view WEMI from purely informational standpoint, since the draft is rather far from even articulating the principles of WEMI in a formal specification.

Technical Administration and Security

Before we move into the developer's view of your CMS, let's start out with key concepts that are central to how you manage content — and the system itself.

Threat Prevalence: How Much Is Your System Targeted?

Packaged software is rarely *inherently* secure or insecure by nature. However, you should understand some important considerations here:

1. The default installations of some packages have known vulnerabilities
2. Some packages are targeted more frequently
3. PHP-based systems tend to be targeted more than alternative platforms

In the end, software doesn't secure itself; people and processes are required. However, the level of effort and attention required will vary from package to package and should be factored into your overall selection decision.

First, let's look at native vulnerabilities. Some people prefer the open source, PHP-based Joomla! Web CMS because it is relatively simple to install and run. However, just as complexity can bring unexpected problems, so too can simplicity. In this case, Joomla!'s

1. Source: www.oasis-open.org/committees/wemi/

default installation has never been considered very secure. Although there are some very nice primers for securing a Joomla! installation, many novice Joomla! installers do not use them.

This in turn makes Joomla! a more target-rich environment, which brings us to our next topic: the most heavily used systems come in for the greatest threats. It's simply more efficient for those with malicious intent. Thus, open source platforms like Drupal and WordPress are frequently targeted.

It's worth remembering, however, that open source projects are extraordinarily open about security vulnerabilities. Finding, publishing, and fixing security vulnerabilities is something the open source community has gotten quite good at, particularly in the Linux world, where every line of code for the entire operating system (including all encryption routines, random-number-generating code, and so on) is available free for the downloading. The point is, someone on your team needs to keep up with the alerts and patches.

On the other hand, it's striking that many of the most commonly targeted platforms have PHP in common. Of course, languages don't create security problems; programmers do. However, PHP's reputation here becomes a bit of a self-fulfilling prophecy, as those systems are perceived as more profitable targets.

In the end, when you install a new system (even just to try it out), and it is facing the public web, be sure to read the security docs first, and when we give a lower rating for "threat prevalence," that means you or your hosting service will need to continue to pay particularly close attention over the life of the system.

Authentication & Authorization: Who's Authorized to Do What

There are two key issues here:

1. Authentication: Are you who you say you are?
2. Authorization: What are you allowed to do?

Most CMS packages will tie into existing corporate directory systems (such as Active Directory or LDAP stores) for authentication, while providing authorization (what some call "entitlements") within the CMS itself, perhaps based on AD/LDAP groups.

Let's deal with authorization first. An enterprise transitioning to a WCM platform needs a system to manage internal access and permissions that is much more robust than that required to support only one or two webmasters. (Access and privileges for site visitors — as opposed to site producers and managers — are addressed later.)

Internal actors can be assigned privileges based on the role they play (the types of things they can do) or the group to which they belong, which defines their authority and, typically, the scope of the content areas they can edit. It is possible (and indeed quite common) for a user to make up a group of one.

Note, however, that many vendors do not have notions of groups in their system, or have groups, but they are really roles. In Figure 4, users can only belong to "groups," but those groups are really roles with set duties.



Figure 4. Managing group entitlements in EPiServer.

Consider the following chart of Groups and Roles for a generic corporation.

Sample Groups	Sample Roles
<ul style="list-style-type: none"> HR Managers – Can only add and modify jobs Product Managers – Can update catalog content only Graphic Designers – Can create and modify image files and templates site wide (may also be a role) Librarians – Manages classification systems and metadata vocabularies 	<ul style="list-style-type: none"> Super user – Performs any function in the system Editor – Approves content Author – Contributes content Metadata Manager – Adds metadata to site content

You may need a system that lets you assign a combination of Groups and Roles. Using the chart, let's say that Nancy serves as an Editor in the Project Manager Group. She approves the additions and updates that Authors have made to catalog content.

Almost all CMS packages ship with generic roles already configured for your use. These products then typically enable you to modify those roles as necessary (Figure 4). However, not all CMS packages allow you to create completely new roles, and among those that do offer this capability, they may not be able to circumscribe functions in exactly the way you would

like. For example, you may want your Metadata Managers to add and modify metadata, but have no other privileges, or for Product Managers to initiate workflow tasks, but not be able to author content. Ask prospective vendors to show you just how to make the roles and groups you think you need.

On the other hand, if you have very simple needs, stay cautious about products that offer highly granular control mechanisms. These can become hard to manage and the novice administrator can accidentally create problems, typically around locking editors out of sections to which they should really have access.

The way authorizations are managed can vary markedly among competing products. For example, some products will let you manage groups, roles, and permissions in the Active Directory, where you are also managing credentials for authentication. (This is not uncommon with .NET systems that leverage Active Directory.)

Among those, some will access the user repository in real time, whereas others require that the LDAP server sync up with (or “cache” credentials within) the product’s *own* internal role and group info on sort of regular schedule. In the former case, you need to make sure that the network connection between your CMS and your directory server is reliable (or should we say, “highly available”). There can be periods where a user whose rights have been expunged in the corporate directory may still have access to CMS privileges, or conversely, has been added to the corporate directory, but is not yet visible to the CMS until the next synchronization.

Note that very few Web CMS tools can access multiple different directories simultaneously; if that’s your requirement, you may need to consolidate or cache credentials at some single intermediate broker or tier.

In general, be clear on how you expect to do authentication (LDAP, Active Directory, system native, or whatever), and be clear on whether user and group permissions are managed in the directory versus the CMS itself, versus both in synchrony.

Access Control & Entitlements Requirements Builder

What level of granularity do you need now, and anticipate needing in the future?

Some packages proscribe certain roles or limit the number of groups you can define. In Figure 4, you can see seven roles “out of the box.” You can theoretically extend these to create new roles, but the level of effort and potential for problems at upgrade time might make it more worthwhile to select a package that could support the number of different roles you need from the outset.

Cloud Services

By “Cloud Services,” we mean to what extent can a particular solution get deployed in a cloud, by you, the vendor, or a third party. It’s actually not a simple feat, but as enterprises seek to decrease infrastructure spending and free up IT resources, the cloud has risen to the forefront on many agendas.

Definitions of cloud may vary and just as importantly, you’ll want to distinguish between Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service

(SaaS). The important thing here is that all three of these models obviate the need to run WCM technology on-premise.

When a vendor says they “do cloud,” it could mean a variety of things:

- **The vendor or one of its partners offer managed hosting on a traditional version of the software in a datacenter somewhere.** You still have control over customizing, extending, and upgrading the software; really, they're just replacing your hardware and network connections with theirs. We could debate whether this is truly “cloud,” but it may be useful.
- **The vendor or one of its partners offers to host your content management and/or content delivery instances in a public cloud service, like Rackspace, Amazon, or Azure.** You'll find many variants here. You could host your content management application on-premise, but take advantage of cloud elasticity for your content delivery infrastructure, to support global delivery or spikes in traffic. Alternatively, you could host your content management application in the cloud, and keep your delivery services on-premise (see SaaS, below).
- **The WCM vendor (or their partner) may or may not manage your relationship with the cloud provider.**
- **The WCM vendor may or may not convert its one-time license fee into a monthly subscription model.** As always, you will need to pay more money to the cloud vendor to achieve greater levels of redundancy, reliability, and global dispersion. Also, don't forget the cost and hassle of VPN connections to your cloud instances.

As above, you are still running “traditional software” as a dedicated instance and are responsible for whatever changes you make to the application, unless the vendor includes that as a managed service. Not all on-premise solutions will work in the cloud today. Most that do so have been certified with only one cloud vendor.

- **The vendor has built a multitenant, SaaS solution from the ground up.** CMS vendors that take this approach include CrownPeak, Limelight (formerly Clickability), OmniUpdate, and — of course — Salesforce. Note that none of these solutions runs in a public cloud, but are hosted in the vendor's own (redundant) infrastructure. Still, they have the benefit of elasticity and monthly billing for hosting and the CMS application itself, with no initial license fees (however, you typically need to pay a setup fee).

With the exception of Salesforce, these vendors also serve as your “integrator,” and perhaps even agency, since they know their platform best and typically don't have many partnerships with outside development firms. They also take care of upgrades, which tend to come in frequent, albeit small bursts. In fact, some customers are surprised to find that they typically don't have a choice of whether — or when — to upgrade.

Nevertheless, be wary of hosting companies or other vendor partners that take on-premise software and convert it to multitenancy to sell as a shared service to more customers; this frequently does not end well.

Now let's turn to some more general cloud considerations. In every case above, you need to address special issues and ensure you have some key elements:

- Sufficient and secure network connections to your remote instances

- Trust in the vendor's security model and procedures — they know this and usually have strong controls in place, but you still want to check
- Awareness of the potential for outages or disruptions caused by other customers
- Understanding of who is going to perform backups and where
- A clear outline of who is responsible for each layer in the stack, including:
 - Network
 - Hardware
 - Operating System
 - Application Server
 - Data
 - WCM Applications: management and delivery
 - Caching and/or CDN
- A clear plan for handling authentication and authorization, including integration with on-premise identity management and SSO systems
- A knowledge of how you will integrate with other enterprise systems where necessary

Bottom line: vendors, consultants, and analysts throw the term “cloud” around loosely. Be sure you know exactly what you're getting when you sign the contract.

System Reporting: Measuring the System, Itself

Both content and experience management beg measurement, and measurement requires metrics, expressed as reports. Ironically, many content management systems don't provide such reports (though many are waking up to the fact that they should), and some of those don't even create logs upon which such reports can be built.

Look back over your business objectives and figure out how your CMS could help you measure how you're doing. You should decide which ones are most important for your business. A small sampling of reports that could help your managers better manage might include:

- Speed of workflow clearance
- Typical bottleneck points
- Duration of content within particular stages
- Who is logging into the system, how often, and who does not use it at all? (See Figure 5 from CrownPeak)
- What volumes and types of content have been classified according to specific nodes in your taxonomy?
- Is there any orphaned content?
- Is there any live content of a particular lifespan that has not been reviewed for accuracy and freshness?

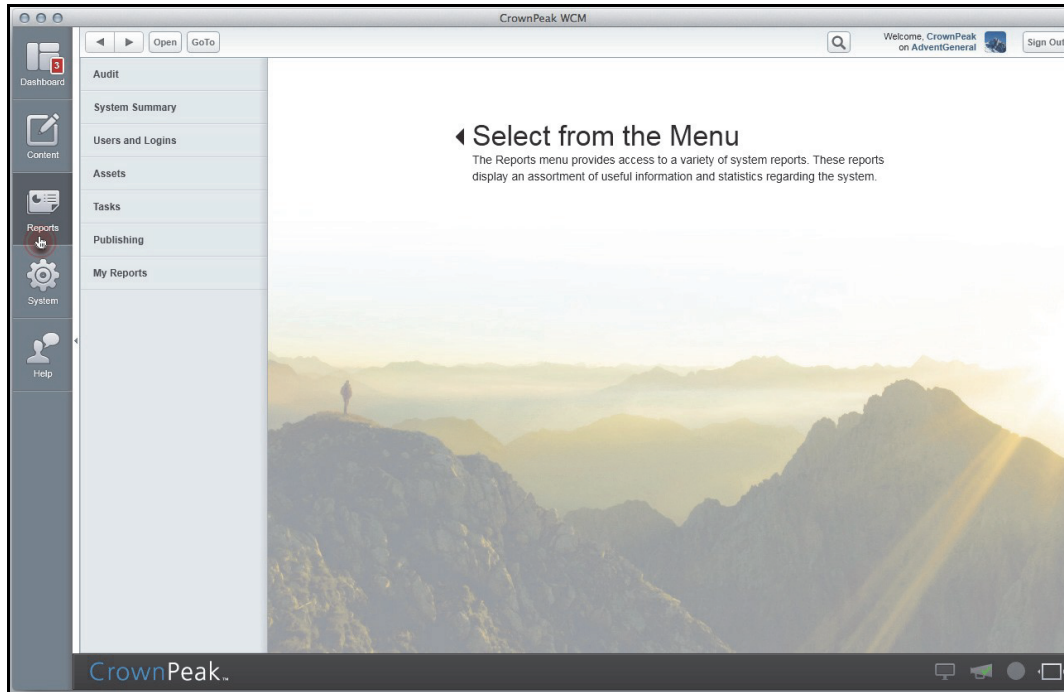


Figure 5. Reporting capabilities in CrownPeak CMS.

Many CMS vendors will tell you that since they provide all the audit data, all you have to do is turn a report writer (like Crystal Reports) against it. That’s a cop-out. You will want reports right within the system itself, so that you can click on content items and tasks to act upon the data you find. You might also want to be able to generate ad hoc reports within the system.

Note that many of these reports can be created by canned search engine queries in the system.

Management Reporting Requirements Builder

What reports will your managers (technical, business, and editorial) require on a regular basis?

List the reports you will want in any RFP and ask the vendors to show you them in their system. Again, start with your business objectives and go from there. If the goal was to reduce your content time to market, then figure out what metrics you’ll need to show that. In addition to business managers, be sure to consult with editorial and technical managers — they’ll be sure to have reports they’ll need to run as well.

Multisite Management: Managing Multiple Properties from One System

Managing multiple web publishing projects from a single CMS sounds quite nice. In theory, you should be able to save time and money by consolidating multiple content management efforts into one. In practice, however, many of these savings can prove illusory. A smart buyer will make sure that key requirements for consolidation are met in any product they select, and will understand clearly in advance which of their departments or websites fall outside the standard requirements set — and therefore remain at greatest risk in any CMS consolidation effort.

The first key challenge in multisite management has to do with user and role management. Distributed enterprises quickly learn that they need particular managerial roles that fall short of “super user” powers, but offer more capabilities than typical authors and editors. In particular, it is nearly essential to devolve basic user management facilities (add a user, assign them to a group, etc.) to designated marketing or content managers, lest a set of centralized user administrators becomes a bottleneck in the entire system. Even if your enterprise uses a centralized directory like LDAP or Active Directory for basic authentication, you almost surely will want to assign specific rights and entitlements in the CMS at a departmental level. Make sure your CMS vendor can accommodate how you want to manage the multitude of users and departments your system will encompass.

The next important challenge has to do with content integrity and reuse. Naturally, many enterprises want to take advantage of a single CMS to reuse authoritative content across multiple web properties. This is reasonable and quite possible, but you will face increasing difficulties as your content becomes more granular, because, for starters, your content analysis exercise must now span multiple departments and uses. Enterprise content reuse therefore implies enterprise-level information architecture. Perhaps more importantly, managing reusable content on an ongoing basis can present greater difficulty than the initial analysis.

Some WCM vendors differentiate between “global” content and “local” content. Make sure you understand how your vendor will allow you to treat localized content editing. Then work through the practical implications at each step of the workflow. It sounds tedious, but will pay off with happier users and more successful content reuse.

Finally, note that many enterprises do not treat this as a technical problem, but fundamentally as a question of user education and human judgment. In this case, editors are asked to investigate the effects of their changes and act accordingly, whether it is spawning a new, localized content item, or putting a global item through an ad hoc workflow for community approval before it is changed, or some other solution.

A final issue in multisite management revolves around deploying content to multiple target delivery environments. We’ll discuss this in greater detail below.

Page Publishing: Where It All Comes Together

At some point, content needs to be married up to templates (see “Development Services: Making Your CMS Your Own” on page 45 for more on content modeling and template development). Many vendors and consultants call this “publishing,” from the content management side to the content delivery environment(s).

Websites that serve static pages from file systems need only a web server. Indeed, web server software is optimized to do just that: Read a file off the disk and serve it up. If you are incorporating other dynamic logic, engaging in any customization, or checking for frequently changing content, pages must be assembled at the time a user calls up the page — that is, at “runtime.” A common expression for generating pages dynamically at runtime is “frying” them. Developers will also sometimes call this “late binding” — that is, you are binding template and content elements on demand, when the visitor clicks, which is indeed late in the cycle.

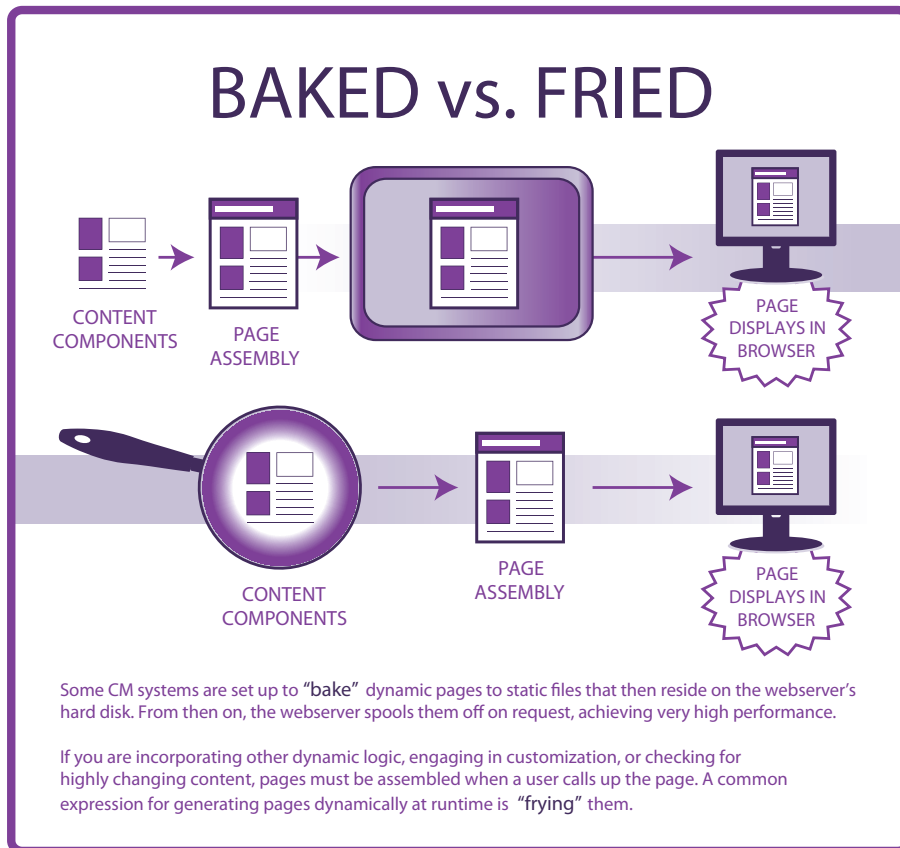


Figure 6. Baking versus frying for page assembly.

The notion of dynamic assembly is not new; it dates to the earliest days of the web with “server-side includes,” where snippets of text from separate files would be inserted into master HTML pages. The technology for frying pages is much more sophisticated now, enabling site owners to pull from different repositories, use conditional logic, and implement dynamic rules to modify default experiences. For this, you typically require an application server of some kind.

That application server does not need to be an expensive, Java-based product. At a basic level, Microsoft’s .NET, and the open source scripting languages (PHP, Perl, Python, Ruby, et al.) can serve as sufficient lightweight application servers in the right containers.

Note that some CMS products still have their own built-in application servers, either homegrown or OEM’d from another vendor, although the industry trend is certainly to use standard third-party application servers or servlet containers, in particular, Apache Tomcat.

Frying brings some disadvantages. It requires a reliable database connection and working code. For performance, it begs a successful caching regimen. “Successful” and “caching” are not words that frequently exist together. Frying pages — especially in lower-end systems — can create highly unfriendly URLs and in some limited cases, erode your public search engine rankings.

Some CMSs can “*bake*” dynamic pages to fully finished static files that then reside on the webserver’s hard disk. Some developers call this “early binding.” From then on, the webserver spools them off on request, achieving very high performance. It also reduces risk of downtime, since static pages do not need to concern themselves with connectivity to a live database. Typically, such page generation routines can be scheduled to run in batches, or individually as separate documents are updated. Obviously, batch-based updates mean that content on the file system may differ from that in the database until the next “baking” session synchronizes the two again.

If you take this approach, make sure that pages that are dependent on an asset that is baked are reassembled and re-baked as well. For example, if you change an image name or the location of a page, other pages referencing those assets also need to be updated and regenerated. “Baking” sounds simple, but in reality can be quite complex, especially with respect to deployment rules, necessary author alerts, and integrity. Indeed, the assembly process is generally quite CPU intensive. If you have to re-bake your entire site (e.g., if your footer changes), it could take hours or days, depending on the number of pages. Finally, the principal disadvantage here is that there is no interactivity; everyone sees the same page all the time (unless you bake out different versions, which carries a lot of overhead).

Some CMS packages support hybrid publishing systems that enable you to bake infrequently changing elements (such as navigation bars), while keeping other page elements dynamic (e.g., those used for personalization). For example, the system may preassemble all the databased elements of a page — except the header — into a JSP file. The header contains some personalization logic that needs to be evaluated at runtime, via a snippet of JSP code. We call this approach “parbaking.” (For more detail see “Performance: Speed and Reliability Matter” on page 53.)

Deployment: Managing the Steps to Go Live

Best practice in developing and managing web properties says that you divide the process into three or four separate but related environments: Development, Testing, and/or Acceptance, and Production (DTAP). The development environment is where you work on code, data, and processes, free from any concerns that the mistakes you will inevitably make will affect your public presence. The “testing” area is where you test your code and iron out any existing bugs. Integrations with other systems also can be tested here.

Your “acceptance” environment should replicate the production environment as much as possible. Upon promoting a version of a site from development via testing to acceptance, you can continue to make revisions on the former while thoroughly testing, and (if necessary) tweak the code in testing and development environments. When a version of content or code passes muster on acceptance, you promote it to production, where it becomes live.

This is especially critical if you are using different systems for Content Services and Experience Services. For example, you’ll need to account for elements that may not be adequately previewable in a development phase, yet still require testing and validation in a non-live environment. You might have embedded scripts or code in your templates that don’t render in your development environment. This case calls for a staging environment that faithfully replicates the production setting, and a final sign-off before promotion to the live server. You can imagine how this becomes even more critical in mobile publishing

environments. Moreover, since in all likelihood you will consistently revise your templates and other underlying scripts and business logic, you will need a mechanism for code promotion, as well.

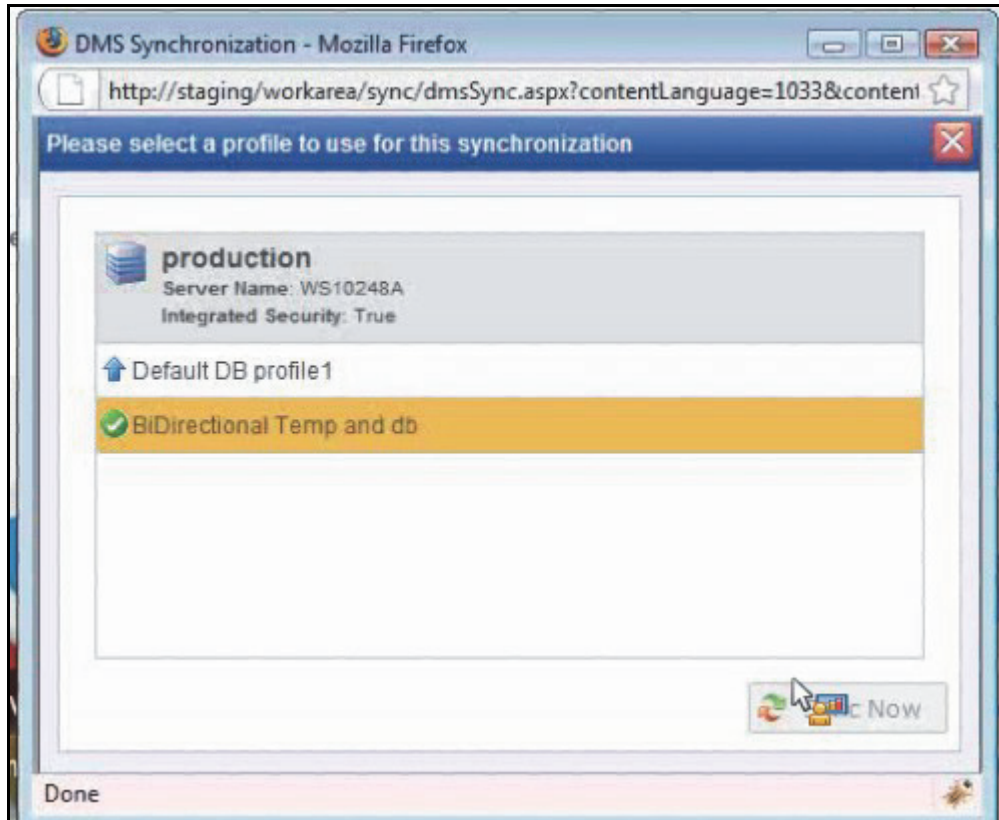


Figure 7. In Ektron CMS, environment synchronization configurations can be saved as “profiles,” which can be applied on-demand to individual sync jobs. The system offers many sync options and provides good reporting capabilities around replication.

At some point, there will be a deployment into production. Even in a tightly bundled management-delivery system, content will be promoted from staging to production. With separate CMS and delivery systems, content is deployed into production over a network. There are various ways of doing this; some packages push you in one direction, and others give you a choice.

Special Issues of Deployment in Large Enterprises

More so than smaller firms and departments, larger, enterprise CMS efforts typically separate content management and delivery more cleanly. Among other reasons, major enterprises have often developed specialized and highly complex delivery applications, especially for e-commerce or other line-of-business applications. Therefore, content deployment — from a management environment to the delivery tier — becomes a higher profile activity on enterprise installations. Moreover, the diversity of CMS repositories, delivery environments, and content types can make for delicate choreography when it comes to promoting content, media, code, and data.

Let’s review some of the synchronization challenges. First, you will probably be pushing data (e.g., metadata records), content (e.g., unstructured HTML snippets, complete web pages, or XML documents), and code (e.g., templates) to the delivery tier, and possibly mapping all of them to a new information model. The data and code need the content and vice versa. For dynamic sites, metadata often drives page assembly from content components or snippets, thus content and data have to remain in sync. However, databases and files systems have similar but distinct ways of updating and synchronizing themselves. Some of your content — even highly unstructured content — might reside in a CMS database, but you can’t avoid dealing with binary files (such as images) in any case. Your CMS package ultimately must be able to choreograph the deployment of both.

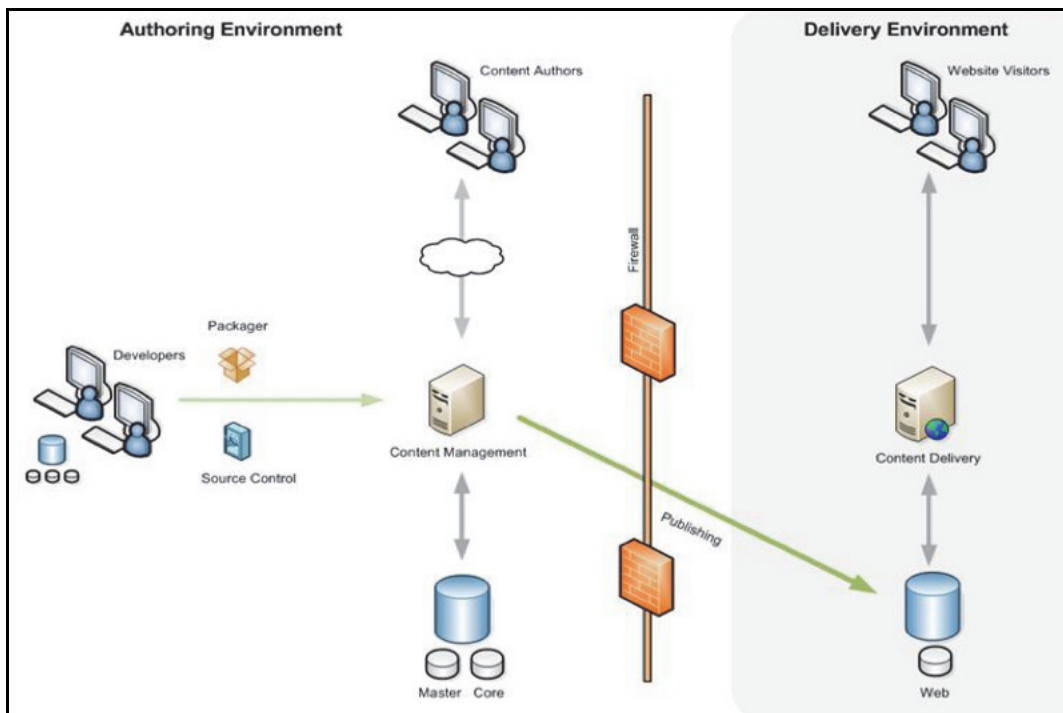


Figure 8. The above illustrates a typical deployment of Sitecore CMS, where the authoring environment is placed behind the firewall, and a delivery environment is in the DMZ.
 Source: Sitecore.

Mid-range CMS packages typically deal with this by pre-generating HTML or employing script files (e.g., .aspx or .jsp pages) that will produce HTML at run time (when a user clicks on a page). In this case, such files are sometimes deployed via simple FTP. Enterprise and upper-tier packages must deal with a broader variety of use cases, and they do so in different ways. However, nearly all of them offer the capability to invoke a combined data and file deployment, albeit using different approaches. Many vendors provide the ability to configure multiple protocols for different deployment targets and scenarios, including HTTP and HTTPS.

When “baking” or pre-publishing are involved, you’ll want to evaluate CPU, disk I/O, and network capacity carefully before proceeding. This can get so complicated that the CMS vendor might want to defer the entire discussion until after you have purchased the product.

Don't agree to this. Involve your architects and system administrators in the analysis and insist that the vendor diagram a recommended approach for moving content from the CMS (likely residing on your internal network) to your testing/acceptance environment (probably in a fairly secure DMZ), to a production, or "live" environment (possibly in a more open DMZ). This will not be a passive exercise for you because your team will need to make some important decisions.

Transactional Deployment

For most enterprises, the deployment of heterogeneous content to and from multiple repositories begs an auditable, transactional system that will ensure and validate that the entire "package" or "edition" deployed successfully, usually through some sort of a two-phase commit process whereby the process concludes only after the entire package has successfully deployed. That way, if one piece fails, the entire transaction is "rolled back" so it can be started anew.

Completing the transaction may require more than simply pushing or pulling content. It might also entail:

- Creating new directories on the delivery server for new content
- "Listening" for a successful data deployment transpiring separately
- Invoking various scripts to synchronize the newly arrived content across multiple servers at the delivery tier

Note, however, that not all vendors can perform all of these tasks. Test carefully.

Development Services: Making Your CMS Your Own

Your Web CMS is an application. You will need to get it to work in your environment. One or more developers will customize it. You will need to manage its configurations; it will go bump in the night. Hosted services obviate some, but not all, of these challenges.

Vendors sometimes equate CMS application development with content modeling and templating — and those are both important (we address them below) — but there is much more to it than that.

Actually, you will potentially manage two or more applications: one that is producer facing, and the other consumer facing. (We discuss this further under "Page Publishing: Where It All Comes Together" on page 40.) All kinds of issues arise here — from the basics of database, appserver, and operating system compatibility — to the more complex, like the frameworks a vendor supports and their approach to application build processes.

The need for application development and management sometimes comes as a surprise to rank-and-file employees, who can use new tools to manage content, but cannot always manage the system itself — whereas previously, they may have held almost complete sway over their desktop and shared files systems.

To be sure, there are various levels of application development in any package. We look specifically at:

- Configuration & Customization

- Extension & Integration
- Content Modeling
- Templating

Configuration & Customization: Making Everyday Changes

These two things sound like the same concept, but actually, they're different. Ideally, configurations can be set and modified by a non-technical user (though as we'll see, that's not always the case).

Configuration means changing some comparatively simple, default settings, often through a browser interface, sometimes via editing a text- or XML-based “config file.” For example, adding and removing users to a system represents a common — and usually simple — configuration. Others can get more complex, and indeed require a developer.

This too may come as a surprise. As web-based tools, CMS products carry a tacit expectation among business managers that they should be modifiable on the fly. Business people worldwide lament that a small change to a content management application must wait six weeks for an inevitably understaffed IT group to invoke a series of tests and code pushes.

Savvy vendors across the content technology spectrum have heard these anguished cries and are increasingly delivering browser-based tools to enable power users and business managers to make structural changes to applications that previously required developer intervention. For example, Web CMS vendors are building forms-based and WYSIWYG tools for content managers to change website structures and interaction rules on the fly. When changing simple parameters (“display 4 related pages instead of 3”), these sorts of configurators become obviously useful. However, a dirty little secret is that today they are frequently complicated enough to remain primarily glorified shortcuts for developers. It is one thing to add a new site directory, but quite another to extend a content type or modify a template used across your website.

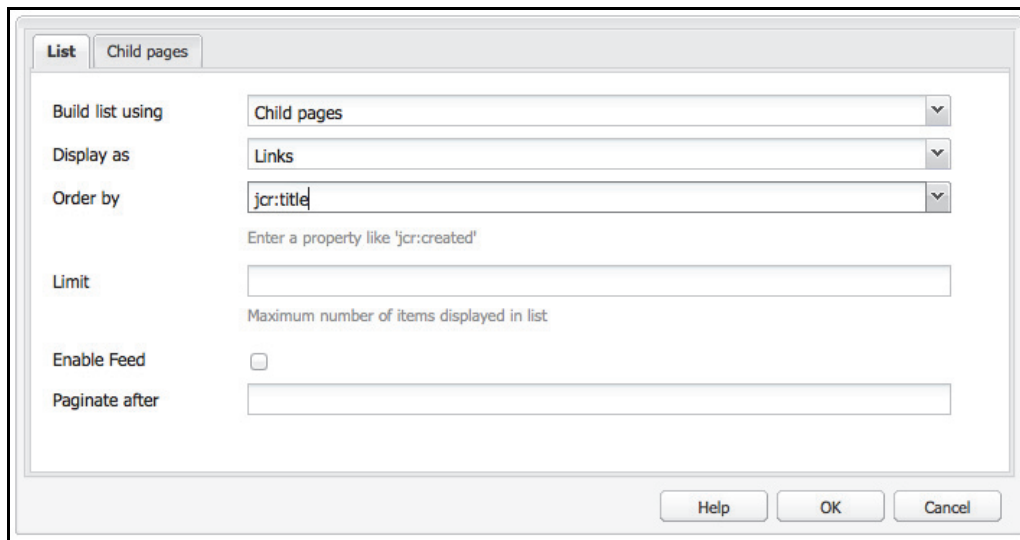


Figure 9. This is a List component in Adobe Experience Manager; where you can configure search criteria for displaying a list.

However, some tools are clearly pushing the line between simple configurations and advanced application development, so you should anticipate some key potential conundrums. First, in many cases, you are changing a production environment of some kind. That immediately raises security and integrity issues. Some (but not all) vendors have responded by establishing configurations as managed objects in the repository so you can version and workflow them. This provides a useful audit trail and perhaps rollback, but does not automatically imply a testing regimen. Additionally, in many modern Web CMS tools, the same configs can be modified in both code and browser interfaces. In fact behind the scenes, some power user controls are really code generators. Clearly, there are lines to draw here.

Note that allowing business managers to change systems will raise governance issues. To the extent that IT was responsible for best practices around “I” as well as “T,” you might lose some important controls along the way. Nevertheless, these configurators are here to stay and can deliver important capabilities in the hands of well-trained power users.

Now let’s turn to customization. Customization usually means writing some custom code, in order to do something that a browser-based configurator couldn’t accomplish. As above, one of your first considerations here should be configuration management.

Just what might need to be customized? You may need to write code to modify: templates, workflow events, publishing behavior, personalization, special contributor interfaces (e.g., task-based interfaces), custom navigation, or metadata-driven display, to name just a few.

Typically, a Web CMS offers a scripting interface against an API. Some tools (especially those in the open source community) have a tendency to emphasize customization over configuration, but customization can offer very powerful and useful capabilities in the hands of a skilled, conscientious developer.

However, customization can also get you into trouble, through poorly written code, memory or security leaks, and thorny problems at upgrade or patch time. Make sure any code is well documented and follows best practices laid out by your vendor. If your vendor doesn’t lay out best practices and you expect to perform a lot of customization, reconsider your choice of tools.

Extension & Integration: For More Serious Development

To “extend” software requires coding, like customization, but typically at a deeper level, through the creation of discrete objects, classes, or modules that run inside or alongside the CMS. You would extend a CMS platform to supply features that might be missing entirely, such as a personalization subsystem, an advanced comparison feature during workflow, a micro-application (such as a survey), or a supplemental archival routine, to name just a few.

All the same caveats we cite above still apply (e.g., the importance of configuration management). However, also you’ll want to pay tighter attention to possible collisions with the CMS itself, as well as potential problems if you muck with lower-level application or platform settings. In the vendor evaluations, when we say a product is more “platform oriented” rather than “product oriented,” we mean it likely lends itself better to extension, potentially at the expense of implementation speed and ease.

Integrating your CMS with other applications may require a combination of configurations, customizations, and extensions, but now you engage in an elaborate dance between two different systems. Typical integrations include CRM, DAM, PIM (Product Information

Management), ERP, and various Digital Marketing tools for tasks like campaign management and social media intelligence.

Perhaps the simplest level here is data integration, integrating information from another repository or system. Many CMS tools now have hooks to draw content from outside their own repository. Of course, even then it can get complicated, especially with respect to security and versioning. You'll want to respect the business logic of both systems when moving or even accessing data, and work through an API wherever possible.

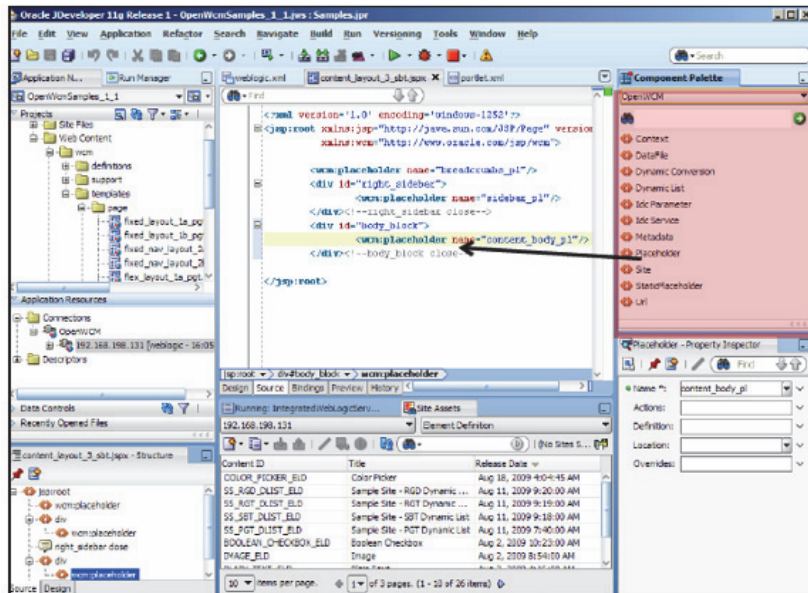


Figure 10. Developing custom pages for Oracle using the JDeveloper plugin.

Process integration is typically more complicated. This is partly because most Web CMS tools do not have sophisticated event models, and because our industry has not standardized as effectively around business process integration approaches. Here you may want to look for supported, pre-packaged connectors between packages, rather than writing your own. For example, many globalization management systems (GMS) vendors offer connectors to bolt onto commercial CMS workflow subsystems, to support handing translation tasks back and forth.

Note that, unlike enterprise portal software, most Web CMS packages were not designed as integration platforms. Therefore, many larger enterprises still look to Portal or Business Process Management (BPM) software as their core information integration platform.

Content Modeling: Manifesting Your Information Architecture

First, let's start with a more basic discussion of content models, before delving into how different CMS tools allow you realize those models in the system.

With the exception of completely free-form web pages, most web content follows a “content model.” Consider the typical press release drafted in a word processor. It contains:

- A logo
- A headline

- Subheads
- A date
- Contact information
- A first paragraph
- The rest of the body
-

Press releases are fairly uniform and familiar documents, and this example is no exception. Note that its structure is fundamentally hierarchical, rather than relational; that is, content pieces flow from a central “element” and have parent-child relationships. For example, the Subhead is a child of the Headline element.

In this instance, “Press Release” is a content “type.” Other content types on your site could be Case Studies, Staff Lists, or Articles. Note that some analysts and vendors use different terms to signify Type and Element (see chart below). It really doesn’t matter which terms your organization uses, provided that you use a common vernacular to communicate document components and structure.



Figure 11. Press releases are typically highly structured.

Content “Type”	Content “Element”
<ul style="list-style-type: none"> •Classes •Objects •Documents •Pages •Templates •Chunks •Archetypes •Models 	<ul style="list-style-type: none"> •Snippets •Objects •Placeholders •Styles •Chunks •Parts/Fragments •Nodes/Fields •Components

So how do you actually create content types? Most systems (though not all) come with some default content types, like “article,” and “blog,” but you’ll almost certainly need to modify those, in addition to creating your own custom content types.

Different systems can take vastly different approaches to content modeling, but in general, they fall into three categories, from easiest to most difficult:

- **Forms-based:** Here a trained administrator or developer uses some sort of forms-based interface to create and define content types and associated elements. In some cases, this will include drag-and-drop interface, especially when paired with a specific development environment (IDE).
- **Configuration files:** Typically, these XML files define content types and relevant elements. The advantage to this approach is that it simplifies hierarchical object modeling — at the expense of complexity. In addition, these files may need to be managed “outside” the system, in a separate configuration management environment.
- **Database field manipulation:** Here, you literally create fields in a database, either directly or indirectly through an API. This is obviously the most developer-intensive, but does provide near complete flexibility.

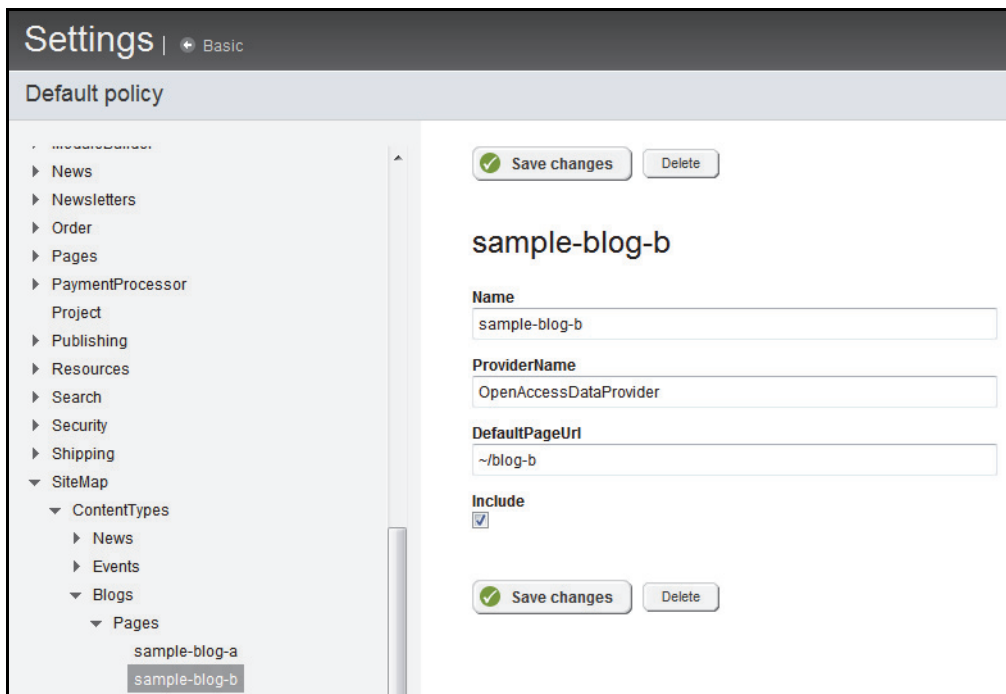


Figure 12. Creating a new content type in Sitefinity CMS.

In general, forms-based approaches work best because you may not need to involve a developer (among other reasons). However, some content types might be too complex for a forms-based interface. Also, remember that content types change, so you’ll want to understand how to modify your model, and not just create it.

Templating Services: Creating Predictable Layouts

At their core, all CMS packages are templating engines. That is, they enable you to publish content out through preset visual and organizational models or “templates.” As an example,

you may have one template for your home page and a different template for your About Us page, since their layouts and designs are likely to vary.

CMS tools vary substantially in how they enable you to create templates, and in larger enterprises, you can expect some sort of debate around this. Some CMS packages provide their own WYSIWYG tools for template creation, which is great for business managers or non-technical designers. Others require you to build templates in an offline editor, such as Dreamweaver, Visual Studio, Eclipse, or their own IDE. This is a step removed from the design process, but allows, in most cases, for proper coding, and enables developers to employ their separate, favored source code control system. Many CMS packages employ a mix of the two processes, sometimes enabling designers to drop in CSS, images, or pre-coded widgets.

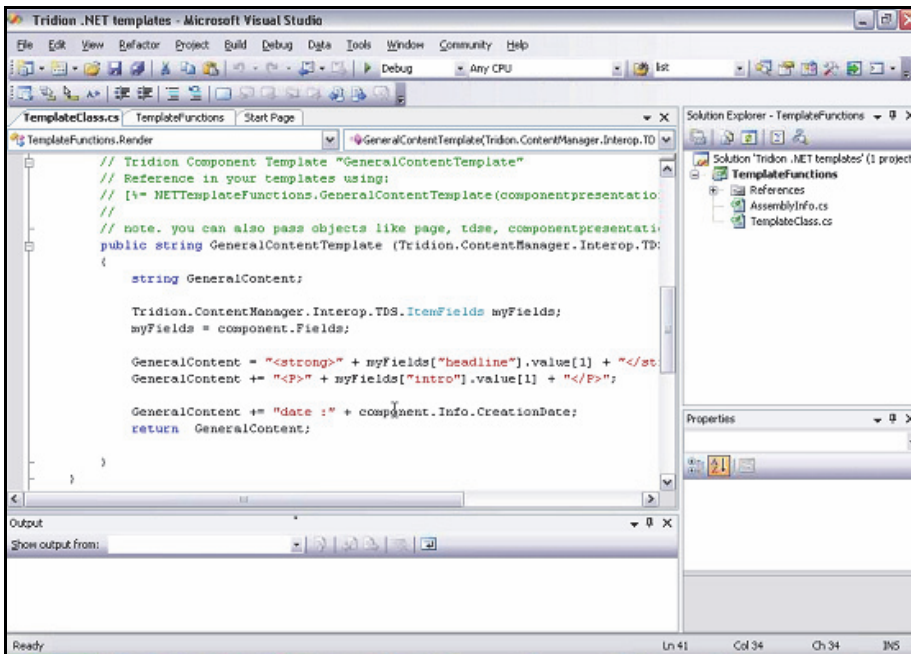


Figure 13. Creating a template in SDL Tridion requires a skilled .NET developer.

A related debate has to do with where the templates are stored. Some CMS vendors believe that templates should be built and stored within the CMS, enabling standard management services such as versioning, workflow, and the ability to be promoted through a lifecycle like any other content item. Other CMS vendors see this as heresy. They believe templating should be left to developers, working on offline files that can be managed in CVS, Subversion, or some other source-control system. This has obvious benefits, but all updates made offline need to be synchronized with the CMS. In either case, it's critical for you to know how you will be creating and updating templates.

```

<meta name="copyright" content="" />
<!-- /com.omniupdate.properties -->
</metadata>
<!-- com.omniupdate.properties --><title>Faculty</title><!-- /com.omniupdate.properties -->
<content>
  <div style="float:right;font-size:10px;padding:5px;margin:5px;">
    <!-- com.omniupdate.div label="photo" group="Everyone" button="hide" --><!-- com.omniupdate.multiedit type="image"
    
    <!-- /com.omniupdate.div -->
  </div>
  Office: <!-- com.omniupdate.div label="office" group="Everyone" button="hide" --><!-- com.omniupdate.multiedit type="text"
  Fax: <!-- com.omniupdate.div label="fax" group="Everyone" button="hide" --><!-- com.omniupdate.multiedit type="text"
  Mobile: <!-- com.omniupdate.div label="mobile" group="Everyone" button="hide" --><!-- com.omniupdate.multiedit type="text"
  <!-- com.omniupdate.div label="location" group="Everyone" button="hide" --><!-- com.omniupdate.multiedit type="text"
  Office Hours: <!-- com.omniupdate.div label="days" group="Everyone" button="hide" --><!-- com.omniupdate.multiedit type="text"
  <!-- com.omniupdate.div label="hours_start" group="Everyone" button="hide" --><!-- com.omniupdate.multiedit type="text"
  <!-- com.omniupdate.div label="hours_end" group="Everyone" button="hide" --><!-- com.omniupdate.multiedit type="text"
  Email: <!-- com.omniupdate.div label="email" group="Everyone" button="hide" --><!-- com.omniupdate.multiedit type="text"
</div>
<h3><!-- com.omniupdate.div label="name" group="Everyone" button="hide" --><!-- com.omniupdate.multiedit type="text" prompt
<!-- com.omniupdate.div label="education" group="Everyone" button="hide" --><!-- com.omniupdate.multiedit type="text"
</h3>Biography</h4>
<!-- com.omniupdate.div label="Biography" group="Everyone" button="hide" --><!-- com.omniupdate.multiedit type="textare
  
```

Figure 14. Templating using a tag library in OmniUpdate. OmniUpdate is one of the few vendors still using its own proprietary scripting language, as opposed to industry standards.

While there are a few vendors that require you to learn their proprietary scripting languages, the majority of vendors have migrated to standard industry tag libraries, which have the added benefit of more universal support. The idea here is the use of industry-standard presentation frameworks (like JSP tagging or XSL templates) to build templates. The screen in Figure 14 is from CMS vendor OmniUpdate. Note that tag libraries (“taglibs”) can be proprietary. This means that you can’t simply take one JSP or XSL template from one CMS and drop it successfully into another one. Nevertheless, you can take advantage of common industry expertise, tools, and development patterns around Java Server Pages technology, and still use standard scripting languages.

In the bad old days, designers frequently complained that Web CMS tools limited their layout choices. Fortunately, those days are mostly passed; template extensibility and component granularity are now key considerations here. Higher-end, more flexible packages will typically allow for very granular control of templates, with a hierarchical model for template development where you can create and manage multiple variants of single template models, and “nest” template elements.

The latter is becoming increasingly important, as sites get larger and enterprises want to control multiple, related properties that may share some design and template elements. Not all systems allow you to nest elements easily from a central palette of template components.

Template Requirements Builder

How many templates do you really need?

To paraphrase Albert Einstein, you should build as many templates as you absolutely need — and not one more. More templates mean more objects to manage, and more to modify when your design changes. If you publish a large site with many divergent templates, be sure to investigate how the CMS allows you manage those templates (and associated elements). Conversely, if your site layout consists of many variants of just one or two templates, you'll want to put careful consideration into the hierarchy of elements, such that you understand the cascading layout effects of modifications to elements higher up in the tree.

Do your designers or developers have decent XSLT skills?

Some XML-based CMS tools use XSLT for templating. This can be inconvenient for your designers, and for more complex projects, could require more than basic XSL skills. Some vendors are transitioning away from XSLT towards open source templating frameworks, like Apache Velocity and Twig (supported by Java and PHP programming languages respectively).

Performance: Speed and Reliability Matter

“The system is slow.” Both site managers and content consumers hear that lament too often. Diagnosing specific bottlenecks requires a holistic view — not least because performance problems can originate on networks and hardware, rather than software — but your CMS can play a decisive role (good or bad) here.

Back-end Performance

By “back-end,” we mean the speed and reliability of the CMS itself for content contributors and managers. As with the “front-end,” much will depend on the amount of technical horsepower you apply to the system, as well as the complexity of the content model and processes you apply.

That said, some WCM tools are inherently slower than others are. The fact that vendors consistently boast of speed improvements should suggest to you that this is a persistent challenge. Those tools that are not easily clustered according to industry-standard practices can present a particular liability here for large deployments.

In general, you should be able to request of any vendor (and then test) metrics for average concurrent (active) users in the system, based on a particular hardware configuration. In some cases, you may find it as low as 20 active users per CPU. Of course, depending on the particular bottleneck, even one lone user can experience a sluggish environment. The key here is not to take the vendor's word for it, but to test before you sign any contracts — especially if any fees are associated with applying additional hardware. In cases where you are evaluating cloud or “managed hosting” for your CMS, perform ample testing to ensure that you do not experience unacceptable levels of latency when accessing an instance of the software hosted possibly in a data center far, far away.

Another big issue here is the “heaviness” of the user interface (UI) code in the application. Vendors increasingly employ greater amounts of JavaScript and CSS to render and add dynamism to their interfaces. Make sure you test these very carefully with subsets of all potential system users. What works well on a developer’s new laptop may prove prohibitively sluggish on an underpowered, standard-issue laptop or tablet environment within your enterprise.

Site Caching & Delivery Performance

To the extent that when a CMS serves as a content delivery service, it is typically serving dynamic pages, in which case you have to consider issues of speed and reliability. Two key issues here are caching and replication/clustering.

Caching

Caching describes a family of approaches to speeding up page delivery by keeping certain information “cached” in the server’s memory or some other readily accessible repository (like the file system — or even the database).

Caching is not just for busy portal sites. Since Web CMS packages tend to be surprisingly voracious resource hogs, caching has become a critical performance consideration even for low-traffic sites that employ a CMS platform. This is yet another reason why many CMS packages have turned over publishing to application servers that employ, among other things, state-of-the-art caching systems.

In considering a CMS package, find out what rules govern any caching it may do, in particular the always-byzantine art of cache invalidation rules. You’ll want to be able to tinker with the settings, depending on how dynamic your content is, and what parts of your site you expect to be heaviest hit. In general, caching can get very complicated very quickly, and you ought to be aware that your CMS supplier may not have a core competency here.

Don’t underestimate the impact of caching on the editorial process, especially with respect to preview, staging tests, and perceived problems in updates when updated content items aren’t flushed quickly enough from the cache.

Replication / Clustering

You may want to replicate your content beyond your core webservers and application servers for the same reason that you cache on those machines: To improve site performance. You improve performance by distributing the traffic load to multiple computers (typically with some sort of load balancing hardware in front of them), and you bypass potential network bottlenecks by replicating to other data centers and networks — thereby getting closer to your end users.

The other reason for replication is to provide some sort of failover in the event of a critical failure on one of your machines.

In the broadest sense, we mean several things when we talk about replication:

- Replicating content to multiple servers at your datacenter
- Replicating across multiple locations (including internationally)
- Replicating, or “caching” different elements or assets at different levels of the broader network.

Load balancing and replication look very elegant in network diagrams, but in real life, they introduce substantial technical complexity to any system. For example, managing user sessions (where the system needs to remember a user from one click to the next) and database connections can get a little tricky with multiple load balancers, web servers, and database servers. Commercial-grade application servers are designed to do just this, but if your CMS is serving as your application server, then you’ll want to investigate how it handles session management behind the scenes. Replicating content repositories brings its own challenges, especially for frequently changing content and multi-directional updates. Bottom line: Look carefully before you leap.

There are licensing considerations here, too (see Figure 15). A CMS vendor will not typically require you to purchase an additional license for failover purposes, as only one instance of the software will be running concurrently. Replication for performance, where multiple servers are running simultaneously, often means that you will incur more licensing fees.

For putting content closer to end users, you have several options for network caching. There are appliance products you can buy and host that do this. You can also contract with “Edge” or “Content Distribution Networks (CDNs),” that provide network caching as a service. CDNs boast that they can integrate with particular CMSs, which is useful — but not essential. Pick a CMS first, then select a network caching service.

CDNs particularly excel in putting images, video, streaming media, and other bandwidth-intensive files on their edge servers. This simple form of replication is indeed very easy to implement. An XML-based format called “Edge Side Includes” (ESI) promises to allow some level of dynamic delivery from caching servers as well. The standard is relatively new, but major vendors are getting involved.

Alternatively, you can distribute the load through any number of different cloud-based services. The issues here are similar to those with more traditional CDNs, including potentially thorny challenges around cache invalidation.

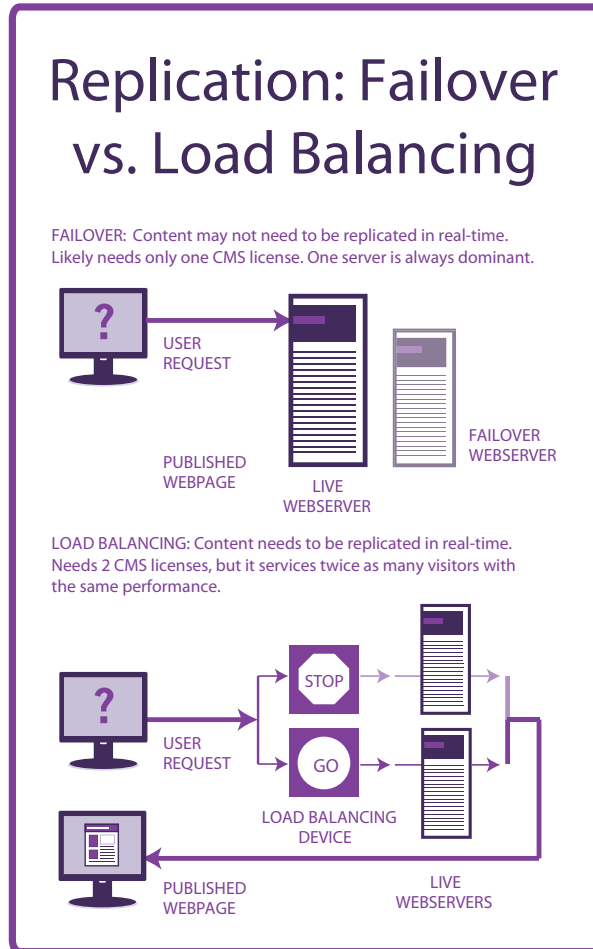


Figure 15. Replication: Failover vs. load balancing.

Scaling & Performance Requirements Builder

How dynamic is your content?

There is a direct tradeoff between performance and freshness. Maximum caching may result in the system serving content that has already been updated. Alternatively, if you always want to serve the latest content on an up-to-the-minute site, the page generation process will experience greater overhead looking for the latest version. Most site owners need something in between.

Scaling & Performance Requirements Builder (Continued)

What will you need to cache?

There are various levels on which a caching system can operate: On files, web pages, objects, or elements. If the CMS uses file-based caching but you store and update your content as discrete elements outside the file system (e.g., in a database), you are not going to see the same performance improvements. The extent to which you are engaging in personalization comes into play here, since you may want to cache certain common page elements (such as images) to compensate for the extra processing involved in assembling a personalized page.

Do you anticipate needing to load balance? At what level?

Load balancing across two or more web servers in front of your content repository offers a simple solution, but does not protect you from network problems to the datacenter or a critical failure in your database. A more robust solution puts your site at multiple data centers. Note however that different WCM solutions will work within this topology in different ways.

How much hardware do you anticipate involving in the final architecture?

One of the reasons you may replicate in the first place is because WCM packages are hardware intensive. However, keep in mind that most enterprise packages (and many departmental offerings) charge by the CPU.

Content Services

Now let's turn to business-facing services.

CMS vendors sometimes call the management side of the system the “creation” phase or “workflow,” although the latter is only a subset of overall content services. Other vendors that focus on this phase use the all-encompassing label of “content management” to describe it, though we consider web content management to encompass both Content and Experience Services.

We define Content Services as everything that happens to content before the end consumer sees it. This is where CMS tools deliver many cost, control, and efficiency gains.

Contributor Experience: How Content Managers Use the System

“What is it like to use that product?” Many would-be CMS users nervously ask this of their peers who have gone before them. This is with good reason; if the experience isn't positive, contributors won't use the system — or use it fully. In this section, we zero in on three key issues: usability, accessibility, and internationalization/localization — all referring to the CMS user interface (UI) itself.

Overall Usability: How Simple Is This Tool for Your Scenarios?

Debate about the usability of content management systems has gone on for as long as there have been CMS tools. Over time, some common norms have emerged and are beginning to propagate throughout the industry, and we'll highlight some of them below. However, the

most important thing to take away from this section is that the most *usable* interface is that which your content managers find most *useful*. The classic definition of usability is “fitness to purpose.” What is logical and simple to one person or company can be non-intuitive for another.

Some user interface (UI) trends are emerging and have spread rapidly across the CMS vendor landscape. Nearly all vendors now provide multiple ways of entering content into the system. After popping new windows became fashionable in the late 1990s, most vendors now keep users to one, perhaps two screens to complete a simple task, using sophisticated JavaScript controls and/or iframes. Other vendors stubbornly adhere to multiple windows — we identify those in the individual product sections.

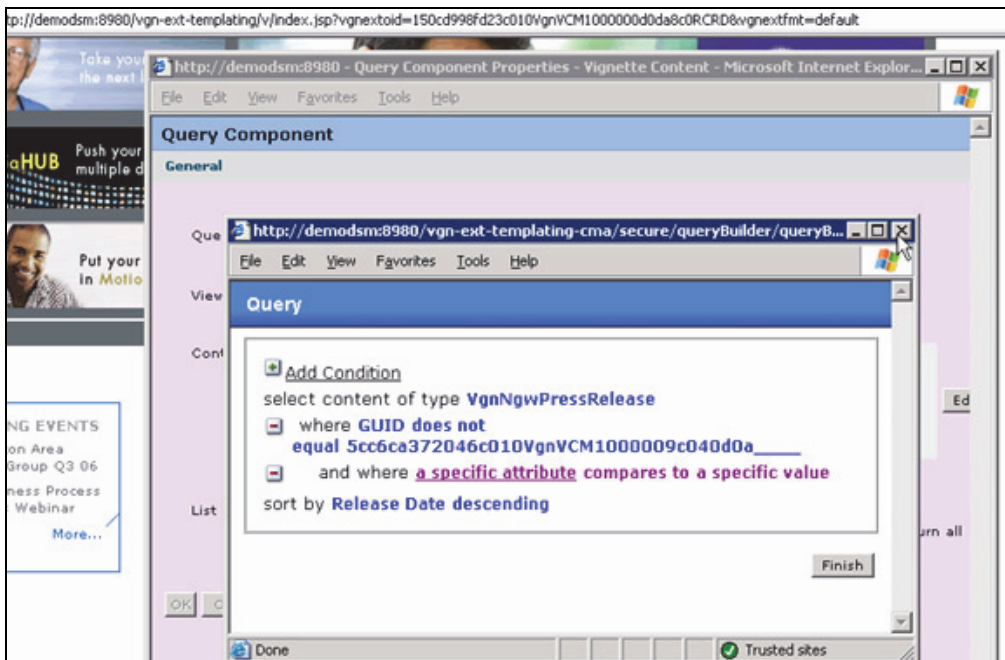


Figure 16. Configuring a dynamic template element in OpenText WEM. Note that three browser windows are open.

Not all vendors offer the same functionality though; even relatively simple functions, such as dragging and dropping items within an interface, auto-completing entry fields, inline status, “hover help,” and wait indicators are missing in some tools.

Many usability issues — such as authoring and workflow interfaces — are addressed separately in those sections. Some other, more global interface topics are addressed here, including:

- User Interface Customization & Adaptation
- Work Queues
- Help Subsystem

Finally, to be clear, in this section we are talking about user interfaces for content contributors, editors, and managers, not end content consumers / customers. For your site visitors, you

control the usability through your templates and navigational structures, although some CMS vendors will circumscribe your flexibility here more than others

User Interface Customization & Adaptation

If the intuitiveness of a user interface remains at least somewhat in the eye of the beholder, then vendors would need to be extraordinarily prescient (or lucky) to know what would work well for your team. One convention the industry has adopted to simplify the discussion is a distinction between “power users” and “casual contributors.” Many CMS vendors offer separate interfaces for these two groups. This distinction can become immediately problematic. Is a power user an author who needs an efficient interface to accomplish the same thing repeatedly and often, or a kind of managing editor who needs a control panel to accomplish a variety of oversight tasks, such as move pages and sections or administer taxonomies? Those two different personas probably will find comfort in very different interfaces. Clearly, the UI assessments that your content managers make of competing vendors is an important consideration. However, in the long run, perhaps equally important is your ability to modify the user interfaces to your particular needs.

Vendors vary substantially in whether and how they make editorial screens available for customization by your developers. In some cases, contributor screens are constructed from XML config files or templates, which are useful for large sites where you may need variants of the same interface but want to maintain some hierarchical integrity among them, but could be limiting due to a lack of a scripting interface. In other cases, vendors allow you to create and modify system screens using the same templating approach and methods that you use to create output templates for content — often some sort of scripting language. In general, we tend to prefer this approach because it makes maximum use of the system’s existing capabilities (e.g., versioning), and leverages developer skills (i.e., learning one templating paradigm, not two). In other cases, vendors allow you to modify input screens via a forms interface; this is the easiest for business people to use to make simple changes, but is likely to be inflexible in the long run and doesn’t always allow for simple UI enhancements (e.g., form field validation). Finally, some vendors — especially at the lower end of the marketplace — do not allow you to modify user interfaces at all. That can severely limit the long-term flexibility of the platform.

We are increasingly seeing vendors providing simpler, task-based interfaces for contributors who only need to undertake a small, consistent set of steps in the system, and for whom the typical control panel screen can be overwhelming. This typically supplements “traditional,” full-blown UIs for power users.

Before you set off to modify any interfaces, however, be sure to get from the vendor in writing some sort of certification about what changes you can make (and where) that won’t expose you to having your work overwritten in any subsequent version of the product. Vendors are increasingly sensitive to this problem and will often put templates in a separate, untouchable directory for just this reason. However, not all of them work that way. Creating your own separate wizards or web application-driven interfaces can compound this challenge.

“Adaptation” is the ability for the editorial and management interface to adapt to specific devices and environments — typically mobile environments. Some vendors have issued native mobile clients for very simple tasks, like workflow approvals. Others have tablet-specific editorial layouts, and still others have modernized their UIs around Responsive Design principles, allowing the interface itself to degrade gracefully, depending on screen size.

Nevertheless, most CMS tools are optimized for desktop authoring environments, at super-large screen resolutions. Test carefully.

Work Queues and Dashboards

Also called a “task list” or “dashboard,” this screen shows what work is in progress in this particular instance of the CMS. Busy users who interact with the system frequently will appreciate the ability to:

- View and sort tasks according to priority and other variables
- View ad hoc tasks as well as content-approval workflows
- See a generous description of the task, including a meaningful title and where that task stands in its overall workflow
- Identify who originated and/or submitted the task
- Take some action on the task without opening up the content item and/or task detail description

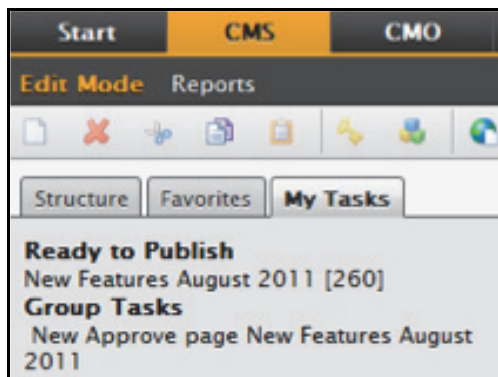


Figure 17. “My Tasks” tab in EPiServer CMS displays both personal and group tasks.

Figure 17 exhibits several (but not all) of these characteristics.

Separately, the system should show which content objects have been “checked-out” by that user, so that they have a quick idea about their own work in progress, as well as know what content items they may be keeping “off limits” to other users.

On the other hand, some casual contributors may want a much simpler and less cluttered in-box that doesn’t tell them things they don’t need to know. Our advice above about being able to modify default screens applies here, as well.

Help Subsystem

Help screens are not useful for everyone. Ideally, your content managers would be so well versed in the interfaces they employ — carefully customized just for their needs — that they would never have to seek any help. In practice, however, users ask for a lot of help, especially in super-distributed management models characterized by an abundance of casual contributors who may use the CMS infrequently. In those cases, “help” often ends up becoming an IT function, in the form of a person on the end of a phone line — even though part of the purpose of implementing a CMS in the first place was to reduce demands on IT staff in the publishing process.

A good help subsystem can provide support on a self-service basis. Ideally, your CMS would have instructions next to each field in a form, or contextual help on each screen in the system. In practice, many CMS packages that offer help content do not offer context-specific help, reducing its usefulness. Still, even generic “RoboHelp” systems are better than nothing, but

remember that your CMS application is likely to be customized in important ways that an out-of-the-box help deck can't anticipate. Therefore, the extent to which you can modify the help subsystem (and make it contextual) becomes an important consideration here.

Usability Requirements Builder

What are the default browsers and platforms used by contributors, as well as default screen resolutions?

Nothing is worse than licensing a product and then realizing that many of its features don't work, or don't work well, for a significant number of contributors and content managers. In particular, investigate and test any system closely in the common browsers and platforms in your organization, and don't forget your designers and creative people who are likely using Macs.

Screen resolution is an important — and oft overlooked — consideration. Some vendors have come out with slick, multi-panel, multi-window interfaces. They look and work great on high-resolution monitors, but will typically not work well (or even fail) at lower resolutions.

In any event, make sure that all vendor demos are set at your enterprise-standard resolution, which you should also specify in any RFI/RFP.

What does “easy to use” and “intuitive” mean to your content contributors?

If you can't answer that, don't request it in your RFP or tender. Be prepared for blank stares from users when you ask, however; most people aren't used to being queried for details here and are likely to respond, “I'll know it when I see it.” That's normal. In fact, many consultants define usability as “the absence of frustration.”

It behooves you to organize hands-on tests of the different systems you consider. Fortunately, an entire discipline of “User-Centered Design” (UCD) has sprung up to help you work through the problem of making your CMS more usable. Learn more about it.

How many users will there be in the system, and how diverse are their roles?

The sheer number of users — along with the diversity of roles — is going to place greater emphasis on your ability to customize the user interfaces, including any help subsystem. If you have more than 100 users, make sure there is context-specific help and some sort of robust scripting interface into all the system UI screens. If you have more than 1000 users, make sure that the help subsystem is editable, and that any internal templates can undergo the same careful “build” and deployment process that new website content areas undergo. With that many users, it's best to roll out “editions” of the CMS.

The more roles you have in the system, the greater demand you'll see for custom interfaces. Some of these can be addressed in the rights management subsystem, which will circumscribe what different roles can see and do, but that is not the same thing as building a usable interface for, say, your corporate librarian.

Finally, before customizing a CMS UI, think carefully if this is really necessary. The majority of advanced vendors now provide a set of interfaces for various roles. You should check and see if those default options will suffice for your needs before investing time and money in UI customizations.

Usability Requirements Builder (Continued)

How much workflow do you anticipate?

If you anticipate only a simple author-approved process for most of your content, and you expect the overall throughput to remain light, then very simple work queues will suffice. However, if you have multi-step workflows, with many actors, and/or high volumes and potentially narrow choke points, then you will rue the day that you foisted a “plain-vanilla” task in-box on your users, as they will likely respond by processing tasks offline, likely using some of the rich queuing and messaging in Outlook and Exchange. Power editors in particular want highly customized and customizable task lists.

Vendors report there is little interest among buyers in customizing help subsystems. However, there is a bit of a “Catch-22” here: When the systems are virtually inaccessible, the information not chunked, and the screens not editable, CMS buyers may avoid the help system, altogether. That means more phone calls to the help desk.

UI Accessibility

With a Web CMS, there are two dimensions to the accessibility challenge: Accessibility of the pages that the CMS generates and visitors consume versus accessibility of the CMS application itself for content contributors and managers behind the firewall. Let’s look at the latter problem first.

Contributor accessibility is not a trivial matter for public enterprises or any organization with content managers with disabilities. By default, most of the native interfaces among the tools in this report are not accessible, although nearly all vendors claim they are working on it. A handful of tools, such as Plone or TERMINALFOUR, are notable in the accessibility of their default contributor interfaces out of the box. Some (but not most) CMS vendors will supply alternate, accessible interfaces. They may not be as feature rich, and you’ll want to test them before signing on.

UI Internationalization / Localization

If you are publishing content across multiple countries or languages your *software itself* may need to support multiple languages and dialects.

This means addressing the administrative and content-contribution screens — especially the default ones that ship with a CMS package — as well as any help screens and error messages. For example, in Figure 18 from Oracle, you can see that the default instructions for in-context editing remain in English and would ideally be localized — along with any “help” screens.

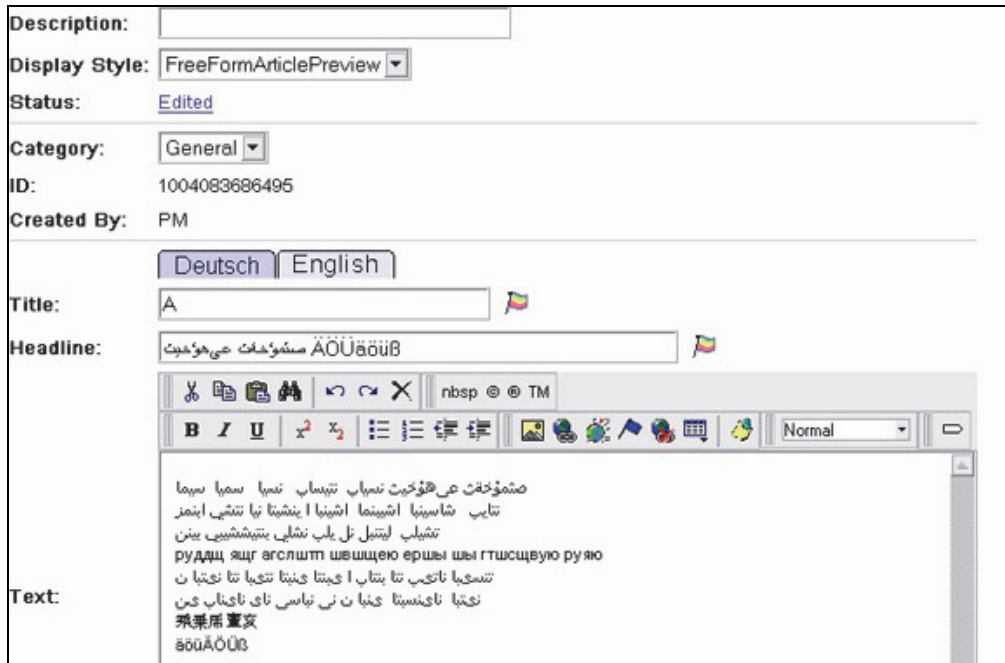


Figure 18. Like most CMS packages today, Oracle supports UNICODE.

From a technology perspective, it is important to distinguish between internationalization and localization. Internationalization is the first step that enables localization. If the CMS is internationalized, it can handle many languages and character sets, and thus it has the ability to *be* localized. A localized CMS ships with various locales or languages already installed. Beware of vendors claiming to be internationalized, but not yet localized, as the burden of localization (translation) will likely fall to you. As you might expect, European vendors tend to provide better facilities for localizing interfaces.

Internationalization Requirements Builder

What languages will you require in the editorial and administrative interfaces of the CMS?

Some CMS vendors have translated their default administrative, editorial and even help screens to non-English languages; others have not. In most packages, these interfaces are primarily templates that are accessible through an API, so you could translate them yourself. However, would you really want to? What happens when it comes time to upgrade to a new version? What if the CMS uses images for navigation or buttons that contain text? How are these translated and handled? If you operate in territories that use languages that are small markets for your CMS vendor, such as Swahili, Czech, or Icelandic, you might be required to handle the translation yourself. Be sure to understand the consequences and costs up front.

Internationalization Requirements Builder (Continued)

For the editorial UI, do you want the language to be set per user or per site?

If your sites are in multiple languages, you might prefer the per-user setting, where German employees can work with the interface in German, and their French colleagues can use French.

Can you get away with only using English characters for your CMS user names?

If you need to have user names such as Jørgen, Jürgen, Noël, René, make sure this is supported even if you plan to integrate with an existing repository that uses non-English characters for user names, such as Active Directory or LDAP. Be sure to test how your CMS handles the logins.

Contributing Content

Most internal CMS users will spend most of their time here, so it's essential to get this right. Your first task when building requirements is to identify these people — the content owners.

Authoring and Transformation: How Content Gets into the System

There are many dimensions to this problem, but let's start with the challenge of finding the page you want to modify, or the location where you want to add a new item. Within a CMS, content contributors need to be able to navigate quickly to the area where they want to add or modify content. This can become an easily overlooked source of frustration. If contributors can't easily find content on their own site that they wish to modify or update, you risk the need to recreate it.

Systems that offer “in-context” editing — where contributors browse to an area of the site in a preview environment, log in to the CMS, and start editing — sometimes substitute this approach in lieu of a more robust retrieval mechanism. Browsing through rendered content is convenient, but may not show you all the content you need to see and dependencies among various pieces of content.

Ideally, your contributors would like to be able to list content by owner, type, and workflow status, as well as list content by individual item and by page — which may be two different things in your system. A critically related service here, then, is repository search (see “Repository Services: Versioning, Version Control, and Repository Search” on page 75).

WYSIWYG Authoring and Editing

WYSIWYG (an acronym for What You See Is What You Get) allows editorial staffers of your CMS to work in an environment where they can see what they're authoring (page layouts, text, and images) similar to how they would display on a final web page. This approach generally makes users feel more comfortable because the environment is familiar and resembles other authoring tools like Microsoft Word, replete with formatting tools, spell check, and other editorial functions, such as preview.

Depending on the WYSIWYG editor embedded in your CMS (and most of them are OEM'd), this way of authoring can be easy or hard. Don't underestimate the amount of training that will be needed, especially for users who don't know HTML. Even such tasks as adding an inline link requires more than basic word-processing skills, and you will probably need to set some corporate policies here.

There are different technical approaches to applying WYSIWYG (or rich-text editing) facilities to browser-based text area boxes. Some vendors may use both types concurrently:

1. **Thick Clients:** These range from Java applets and ActiveX controls to full-blown Flex clients. Their chief drawback is that they are typically proprietary, which makes them difficult to modify. They also create an additional support burden, and indeed may not even be allowed in your standard desktop environment
2. **Browser-based, (JavaScript-oriented) code:** This is generally the most lightweight approach. Beware of under-tested code, though, as it can crash your browser. In addition, complex scripting can increase load times — not always a lot, but often enough to be perceptible to users. The key here is that typically the interface doesn't work until *all* the code has loaded, and in fact, can throw an error if a user tries to access a function whose accompanying behaviors have not yet loaded. As always, test before you buy.

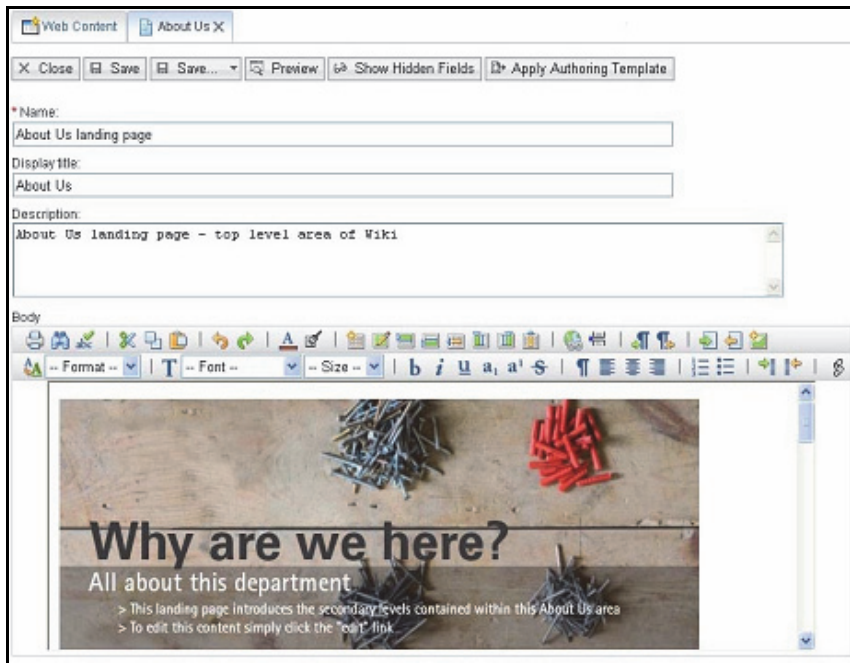


Figure 19. Typical rich-text editor (in this case, the Ephox Java-based control).

Note that with both approaches, you are *auto-generating* HTML tags, although a savvy user can always view the source. Experienced interface developers know that this brings advantages and drawbacks. The principal advantages are ease of use for non-technical contributors, as well as a kind of enforced coding standard. The main drawbacks revolve around the quality of the code, and the occasional need to make adjustments at the code level. Many of these widgets have accessibility problems and likewise the code they generate may fail the more strict accessibility screens.

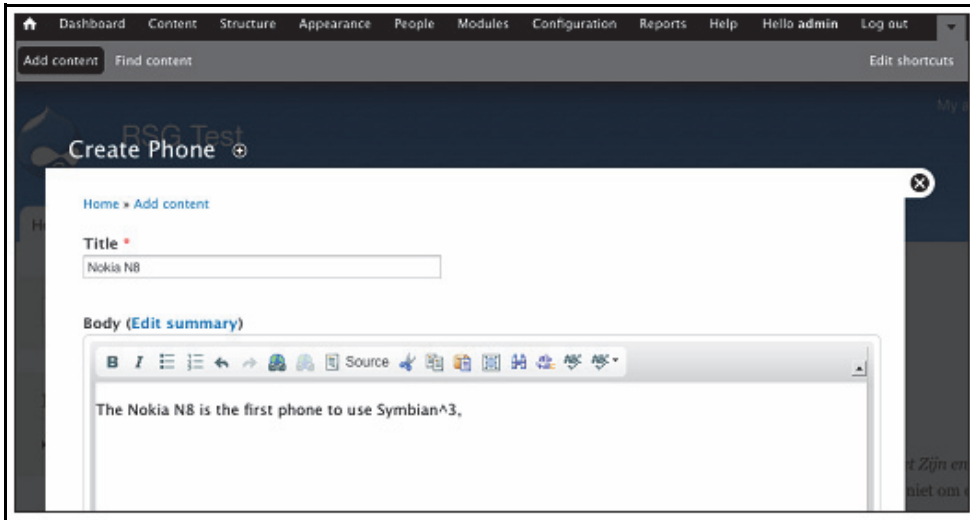


Figure 20. A lighter, more stripped-down rich-text editor (CKEditor in Drupal).

Most authors, in most implementations, in most enterprises, don't like the rich text editor that comes with their CMS. They are notoriously buggy (or perceived as buggy by non-technical users), and inevitably don't deal well with more complex formatting, like nested tables or multilevel lists. The bigger problem is that no rich text editor is truly WYSIWYG from the author's standpoint. For example, your CMS might automatically clean up Word-generated HTML content. However, a potential problem may arise when the cleaned up HTML then appears very different to the author, who could be confused why the now seemingly quite unusable system is messing with formatting.

Nevertheless, it is always a good idea to ensure that the system enforces "clean HTML" requirements (no unclosed or improperly ordered tags, no mixing of uppercase and lowercase element names, no use of deprecated tags, etc.) on any formatted content that is written to the repository. There are various ways of doing this, usually entailing the use of HTMLTidy (an open source tool) or some other filter on the server.

Transforming Desktop Content

If contributors use a desktop package — like MS Word — to develop and edit authoritative versions of source content, the CMS would ideally convert those documents quickly, ideally without manual copy and pasting. As a practical matter, though, many CMS users *do* end up copy pasting, because the content in the Word file doesn't match the structure that's expected, or because they can control the formatting better than any automated tool does — or both.

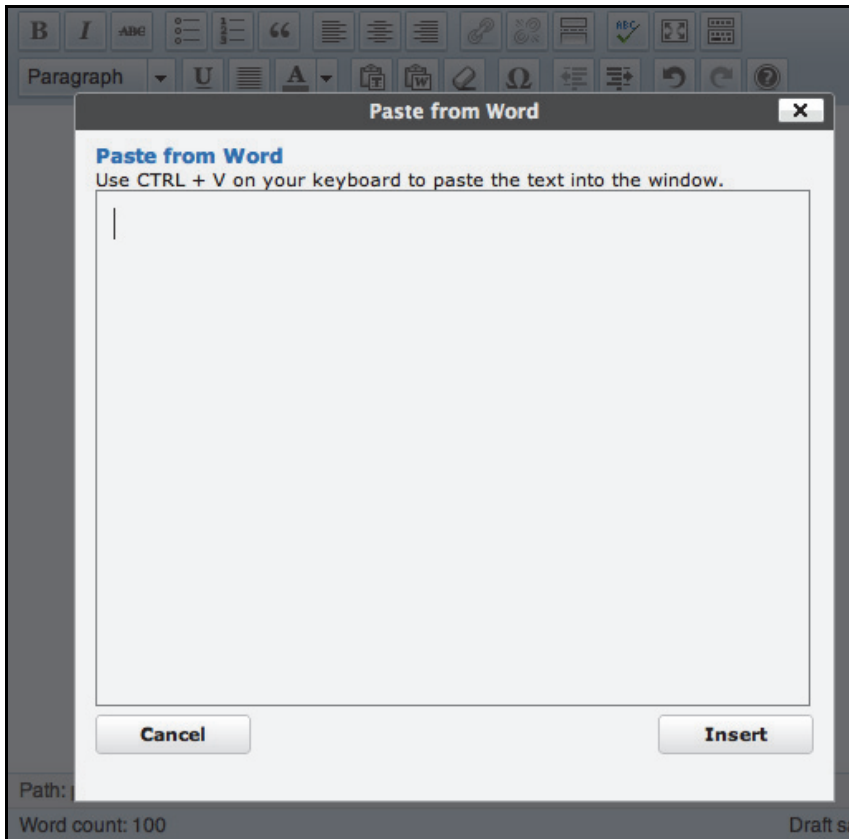


Figure 21. The “Paste from Word” function in WordPress is an alternative to “Paste as Plain Text,” but almost any choice you make could lead to unexpected markup or appearance.

Converting documents demands some sort of automated file transformation (to HTML, XML, or plain text) and import into a content repository. CMS vendors typically deal with this in one of two ways:

1. **Via plugin to the specific desktop application.** These plugins are convenient because they connect the desktop application to the CMS, but they remain rare.
2. **Via file explorer.** Users drag and drop files into the CMS from their desktop (see “File Drop” on page 68).

Conversion tools have become rather sophisticated, and WCM vendors who haven’t built their own are quickly to moving to license such transformation technologies from others. *The often unspoken assumption, however, is that the source documents are structured — and structured consistently.* For word processing programs, this means users must adhere rigorously to strict style sheets or templates that delineate both the various elements within a document, as well as their relationship to each other. Free-form documents offer no cues to a software tool that must transform the text to another format. Documents with inconsistent structures will fail to import, or worse, fail to import correctly.

Converting authors to templates or style sheets within Word or any other desktop application presents significant training challenges, at best, and severe change management problems, at worst. (Vendors usually don’t tell you that in their demos, but experience suggests it is quite true.)

In-Context Authoring and Editing

Most (but not all) CMS vendors now offer “in-context” editing, where, after logging in, you browse through a version of your site in the system until you get to the page you wish to edit. Click on the “Edit” button, and either a forms-based editing window pops up or, in some cases, text can be edited right there in the same window.

This kind of approach is useful for casual contributors who just need to make quick changes — e.g., the archetypal administrative assistant who has to keep the intranet phone list up to date — but who doesn’t need and doesn’t want to do more complex tasks in a CMS.

In more powerful systems, you can even rearrange elements on the page or add new elements — if your content model allows. However, for content that is being repurposed, you will want to insist on a preview of all the different renditions scheduled for that document. For content being reused, in-context editing presents an even greater challenge, because the editor is working on text in a specific presentation environment when in fact that content is supposed to be presentation neutral. This doesn’t have to be a major problem so long as your CMS package can report on dependencies in the system in a useful way (many can’t), and content updaters have been trained properly to recognize the downstream implications of their changes.

Note that in-context editing does not always come “out of the box,” especially in CMS packages that do not deliver your content. Additionally, some systems support in-context editing using a staging instance, which means that authors may not be seeing collateral changes that other contributors have made but have not yet promoted. Nevertheless, these hassles may be well worth it if the system becomes more usable for casual contributors.

File Drop

Many CMS tools enable you to add content to a repository via drag-and-drop functionality from folder to folder on your desktop. Ideally, they do this using the open WebDAV standard (see discussion above). Some vendors use special client programs that emulate WebDAV, but provide a richer set of services (such as popping a window to apply metadata, or showing a repository in a richer way than a set of folders in Explorer).

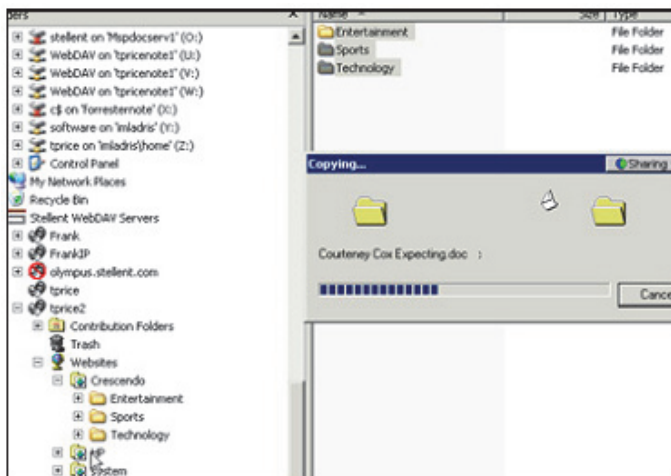


Figure 22. Drag-and-drop functionality via WebDAV.

This approach is limiting to the extent you are just transferring files (rather than more granular content), and you may be limited in the kind of classification you can apply (usually just folder name), but it is very handy for putting multiple files into a repository at one time.

Tagging & Taxonomy: Adding Value to Content

Sometimes lumped under the general category of “metadata,” this technically concerns building, managing, and applying content classification systems. When users — or systems — apply metadata, they are helping to liberate the underlying value of your information.

The first major benefit to tagging content is that it helps people find things. Metadata enables your search facilities to become more targeted and efficient. How so? Imagine your library with no catalog system, no labels on the shelves, and books without title and author info on their spines. Finding the specific information you seek would be extraordinarily difficult. Metadata enables you to provide greater meaning and cues, without which content is just text.

A second major benefit to tagging is establishing relationships. Content elements or documents can relate to each other in time (“sort by date”), subject (“show me both the 3-D image and detailed specs on the product line I’m viewing right now”), and other attributes. Tagging enables this.

Assigning metadata is often seen as the job of content contributors and editors — who else can better apply meaning to their content? Of course, some metadata can be applied implicitly, or transferred from a source document. This includes such attributes as date, file type (where relevant) and size, author, approver, and so forth. The availability of this new implicit metadata is a powerful reason to implement a CMS in the first place.

However, because entering explicit metadata requires extra manual effort, people generally don’t like to do it. If tagging requirements are particularly onerous, contributors will rebel and either put in bad data or find workarounds to avoid this work altogether. Most WCM systems enable you to apply rigid technical controls to prevent this, but a better strategy revolves around getting buy-in from contributors for the extra work. Point out to your contributors that good tagging likely means that more site visitors will see their content, and that since the contributor herself will find the text more readily in the future, updates will come easier. Explain that metadata is critical to whatever business objectives you are pursuing by linking content to core products and services that pay the rent.

Another school of thought suggests that you should give up on getting good explicit metadata from busy authors, and turn this librarian function over to skilled information specialists. Others argue for adopting “auto-categorization” software, which attempts to discern metadata from a document or page automatically. Usually offered by search engine or other third-party suppliers, these tools can raise many expectations internally, but we counsel some caution here. Auto-classification tools need to be carefully tuned against an existing corpus of content and tweaked on an ongoing basis. They can provide a false sense of confidence and generate many false positives if authors are not alert. It’s best to use them for auto-*suggestion* rather than fully automated tagging.

Taxonomies

Information architects or others with librarianship training will add value here by either developing enterprise-wide vocabularies or applying industry-specific classification schemes,

called “taxonomies.” Fortunately, various automation tools have matured to the point where they can speed such things up, but ultimately, this process needs human oversight.

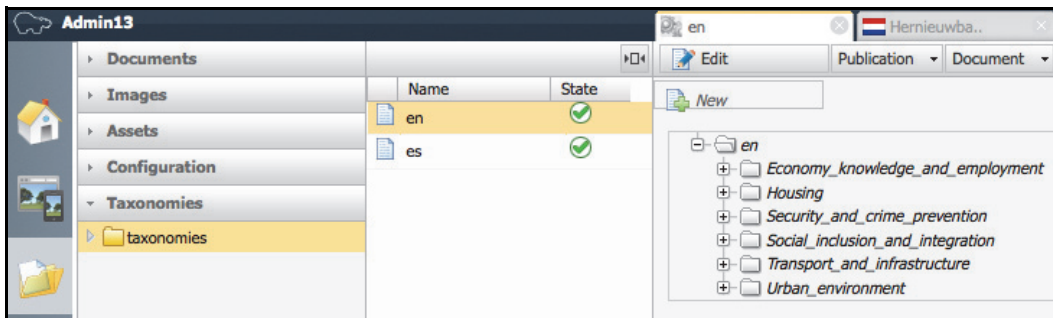


Figure 23. Managing a hierarchical classification in Hippo CMS.

Taxonomies also should be built with content consumers in mind. In dynamic, content-rich websites, metadata may significantly drive visitor navigation and searching. If you want user-centric site maps and navigation schemes, then you’ll need to apply vocabularies that are familiar to your visitors.



Figure 24. A part of the UK “Integrated Public Sector Vocabulary.”

Our final advice is to avoid making contributors do too much tagging at the start, and keep your classifications schemes simple. Yes, metadata provides an important value within the system, but like workflow, it can be overdone. Keep in mind that you will need to update your vocabularies persistently as content changes, and good contributor feedback mechanisms will be essential to keeping taxonomies relevant. The important thing is to do it in the first place — especially if your website exceeds (or is going to exceed) 1,000 pages. Without metadata, you won’t be able to keep track of all that content.

Note that in some industries and public sectors in some countries, standard taxonomies have emerged. For example, all UK government entities must support the “Integrated Public Sector Vocabulary.” Among other things, this prompted CMS vendors in that region to upgrade their tools to support hierarchical classifications — or miss out on that important marketplace.

In organizations where taxonomies are especially critical (such as news sites and scientific/technical intranets), a taxonomy guru typically emerges who serves as a kind of “context administrator” for the system at large. With this authority in place, some firms then sometimes find it easier to offload some or all metatagging to interns or other support staff, properly overseen by the head information architect.

Tagging Requirements Builder

Who and what can have access to metadata and vocabularies?

Your search engine should certainly be tuned to leverage any metadata. Will you need your internal reporting systems to tap into it too? For example, do you need to know how many times a given tag has been applied? How easy is it to update vocabularies in the tool you are considering? With iteration and learning comes revision, so changing the overall taxonomy and individual vocabularies should be as simple as possible — ideally using the CMS itself to manage versions and approvals.

At what level do you want to tag content?

It is theoretically possible to apply metadata to content at any level, including folders, documents, content elements, even individual words? As with everything else, the more granular your system, the more complexity you introduce. Multiple levels of metadata can also have performance implications for the system at large. Your business objectives and workflow should guide where you fall on this spectrum.

Content Reuse

Reusing content pivots on structure. A structure is essentially a model for an archetype. Structure offers the only way to “liberate” information that is otherwise locked up in unstructured HTML files and documents. However, this implies a mind shift at the author level. Content owners need to begin to think in terms of content elements, rather than entire documents.

Many documents are intrinsically quite structured — yet content owners have not usually had to think of them that way (except perhaps for presentation consistency).

The practice of analyzing your content and breaking it apart into discrete, reusable pieces is known as “chunking.” Chunking is the key to a more object-oriented content strategy. Elements can take on object-like properties, such as inheritance, that allow you to extend your content model in a more organized way. Perhaps more importantly, an object-oriented approach allows you to separate content not just from *presentation*, but from *location* as well. Instead of affixing a piece of content to a particular place in your website, you assign it various properties that determine where it will show up in your site hierarchy, and establish appropriate lateral and vertical relationships automatically. *The ability to do this is an important differentiator between page-based and component-based CMS packages.*

Some documents are inherently unstructured, or too dissimilar from other documents in their class or group. These can still be converted to HTML, or very high-level XML, and stored in that format. Some CMSs save documents in their native format (but put metadata into databases), or store documents as “BLOBs” (Binary Large Objects) in a database. At the same time, even very unstructured documents can be effectively repurposed. Read on.

Reuse versus Repurposing

Thus far, we have been talking about *reuse* in a very general way, but actually, we should be more precise. Strictly speaking, reuse means taking authoritative information fragments that live outside of any specific visual context, and publishing them in different combinations to different locales. This might include, for example, product content that might be published in one form on the web, but then combined with specific case studies for a print brochure.

A simpler way to obtain more value from content is by *repurposing* it¹. Here you convert a single document to multiple formats, possibly stripping and/or converting some elements for certain media types. At the simplest level, most content management packages can repurpose Word content to HTML. At a more complex level, you may have XML-based documents that you wish to transform into more than one output: e.g., one rendition for your website, another for mobile devices, and another for a syndication feed. In this case, the transformation occurs on the entire document, and there is no notion of assembly of atomic units. All renditions of the source are visual renditions, as opposed to structural renditions.

The distinction is important, because nearly all content management vendors suggest that they support atomic reuse, but most of them, including most of the XML-based web content management players, actually just support repurposing. Indeed, for most of their clients, repurposing is the goal, and can bring value to an enterprise with multiple delivery challenges. However, another reason why you see more content repurposing in the web arena than content reuse is because the latter is actually quite difficult. A solid reuse strategy requires substantial up-front analysis, a more complex content management system, but perhaps most importantly, it requires a major cultural change among content authors and managers to work on content snippets, rather than entire pages and documents.

Authoring / Transformation Requirements Builder

Do your contributors all know HTML (or XML)?

Typically, the answer to this is “no,” so most WCM packages include WYSIWYG content formatting tools that support forms-based entry and editing. A question here, though is whether you actually want your contributors formatting their text. Check to see if the interface supports style sheets. This way you can control the look and feel of your presentation globally and override the marketing intern who likes to add in-line emphasis through red, 16-point Helvetica Narrow type.

1. Our community is indebted to Ann Rockley (www.rockley.com) for clarifying this distinction.

Authoring / Transformation Requirements Builder (Continued)

Will you need to convert documents — including batch import?

If you are regularly converting large documents, you will want to look into batch import functionality. This is less important if you principally need to perform a one-time load of legacy documents.

How much link validation will you need?

Validating that your internal links all still work is a tricky undertaking. It becomes more important in editorial models, where authors need to emplace many inline links into text paragraphs, as opposed to relying on the system itself to generate and track “related” links in a separate element. Some systems address link integrity at production time, others through periodic sweeps, others at run time, and still others not at all. Note that resolving links can be a CPU-intensive process, which can hinder your publishing or delivery systems, whichever is looking up link IDs and converting them into actual hyperlinks.

Media and Document Management: Files in Your CMS

Managing and publishing files from your Web CMS can get a bit confusing:

- On the one hand, your CMS may have generic abilities to handle files
- On the other hand, the way you want to manage documents (like Word and PDF files) could be quite different from the way you want to manage media (like video, audio, and image files)

Documents tend to be more prevalent on intranets, while multimedia becomes more important on public sites — but of course, that can vary. Documents tend to be “attached” to pages, while media tends to get embedded within pages.

Let’s start with documents. Don’t assume that your Web CMS is a document management system with sophisticated workflow, archiving, collaborative editing, and document conversion capabilities. That said, your WCMS can:

- Probably store documents (in the database or on a file system with a pointer)
- Perhaps enable you to add some metadata to them
- Likely let you reference them from your pages as objects so that if they move, their links adjust accordingly

Digital and media asset management is a somewhat more complicated service. In the CMS Product Comparisons section, we pay close attention to asset management, long the Achilles’ heel of CMS packages. Many CMS tools treat images and other media as generic binary files, such as PDFs, when in fact, images and media represent an increasingly important part of web content — and therefore need to be managed as such. Aggregating media and text content is therefore a key challenge facing any CMS project.

To manage media on web pages, editors usually need (at a bare minimum) access to granular media libraries, with images sorted by type or other attributes, all according to appropriate access rights. Most CMS packages today will let contributors peruse thumbnails and manually add images to pages. A CMS would ideally store key metadata about each image (such as alt

tags, height and width, perhaps even origins and rights), but as a practical matter few do. Some CMS vendors will point you to the new DAM product they acquired, but be careful here; you may risk paying US \$100–200,000 more in licensing fees for the privilege of managing your media content in a browser.

Many CMS vendors now provide the ability to edit images on the fly in the browser. In some cases, the core technology is ActiveX; in other cases it's Java applets; and in still others it's a combination of JavaScript on the client and server-side routines that manipulate the bits and send them back to the browser in real time.

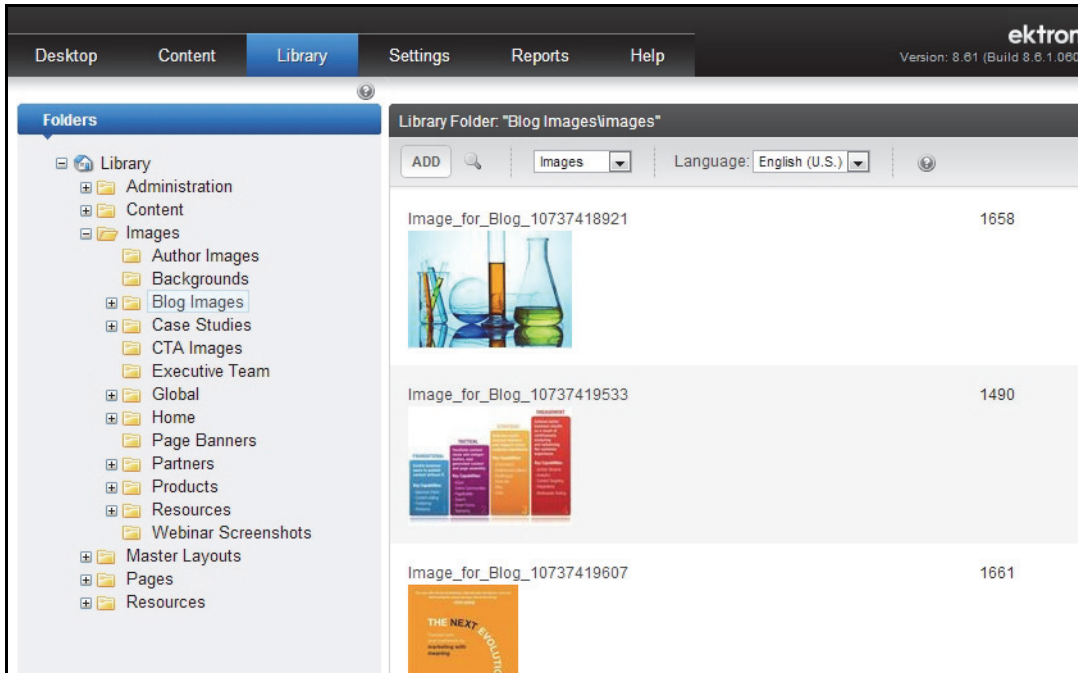


Figure 25. Image library in Ektron CMS.

This capability is a double-edged sword, however. You may want to allow a marketing person to crop an image or modify its size, and perhaps even overlay some caption text. There are other capabilities — such as blurring/sharpening images or changing color depth — that might be best left to a designer.

Note that these rich media editor widgets are fairly new and may prove to be a bit buggier than the rich text editors they are emulating.

WCM tools can increasingly handle image and video “collections,” as well as variable “renditions” of a single image. Asset collections are just that: logical groupings of media files that you can display together, say in a slideshow. Most (but not all) CMS tools have a notion of asset collections, even if they are just sub-folders in a larger asset repository.

Renditioning is trickier, and it hinges on the system having a proper object model around assets. In WCM platforms that support renditions, you typically upload an uncompressed version of the image, and the system automatically creates various renditions of that image for all of the key placeholder sizes that you designate in advance, which can also include a mobile rendition. When the image displays on the page, the template knows which rendition to insert. Nice, huh? The tricky thing here, though, is whether you’ll want a different crop for certain

sizes. Some CMS tools actually support that sort of manual intervention, via the above-mentioned image editing widget.

Finally, you might want to integrate with an external video delivery system, like Brightcove or YouTube. It's simple enough to copy their embeddable code to drop in the appropriate player and video, but more advanced WCM tools will offer more here. Specifically, with the right credentials they can remotely access your Brightcove or YouTube library and enable you to drag and drop video assets onto pages from within the CMS, while setting key configurations (like size) in a forms-based interface. That's handy.

Repository Services: Versioning, Version Control, and Repository Search

This is where you control your content repository/ies. “Repository Services” (sometimes called “Library Services”) sounds mundane, but is essential for a well-functioning system, and may be the biggest immediate improvement for your internal contributors over any ad hoc system you have today.

There are really three pieces to the puzzle here:

1. Version Control and Check-in/Check-out
2. Versioning
3. Repository Search

Version Control and Check-in/Check-out

“Version control” is shorthand for a system that keeps people from stomping on each other's work in collaborative environments. This is typically enforced through content check-in / check-out facilities that prevent two or more people from working simultaneously and unknowingly on the same item.

CMS users in distributed publishing systems often find version control tremendously helpful, but it sometimes can come at the expense of collaboration. The idea that only one person can have a content item checked out at any one time can lead to very linear processes and sometimes added stress on workflow routines (such as excessive looping). Some CMS vendors have responded with specialized capabilities for multiple editors to work on a content item simultaneously — typically by “saving” rather than “submitting” — before promoting it to the next step in a workflow.

Versioning

A related concept is the ability to publish versions and “editions.” For example, you might want to be able to publish different versions of your site to different audiences or separate URLs. This kind of mass customization allows you to circumvent the expense of personalization.

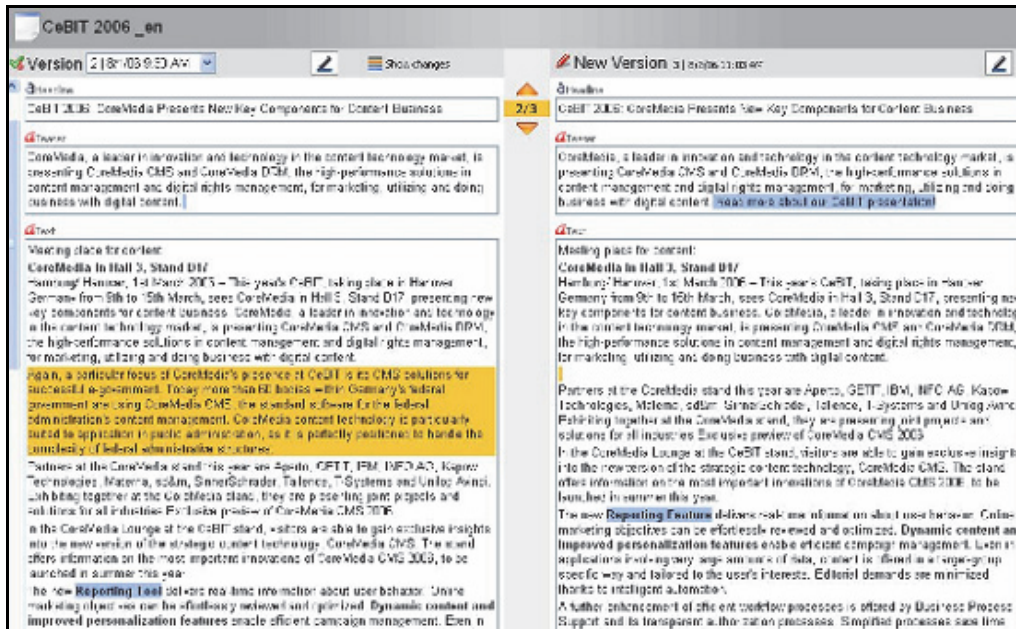


Figure 26. Using version compare to see changes made to content. Not exactly like Microsoft Word’s “Track Changes,” but still visible.

Versioning also enables you to track changes, so you can audit and monitor what changes have been made to any element or asset. Some packages have done this in a sophisticated way relative to desktop tools. Other packages perform change tracking in a simpler, side-by-side way that can also be useful. Contrast this with Figure 27, from HP TeamSite, where the two versions are “virtualized” side by side — very convenient visually, although the changes between the two versions are not highlighted.

“Versioning” has also become a major liability-related feature, and vendors have honed in on this as a critical selling point. As a company, you are responsible for what you put online, but whereas print content may be frozen in time, web content is dynamic, so how can you track what your site actually said at a particular date and time in the past?

Within the ever-litigious US environment, versioning has emerged as a huge issue. For legal reasons, even companies in largely unregulated industries have to be able to recall exactly what their website was publishing at any given instant. The ability to “roll back” a website to that time is therefore essential, and this is another rationale for making a CMS investment has emerged: It’s a necessary insurance policy.

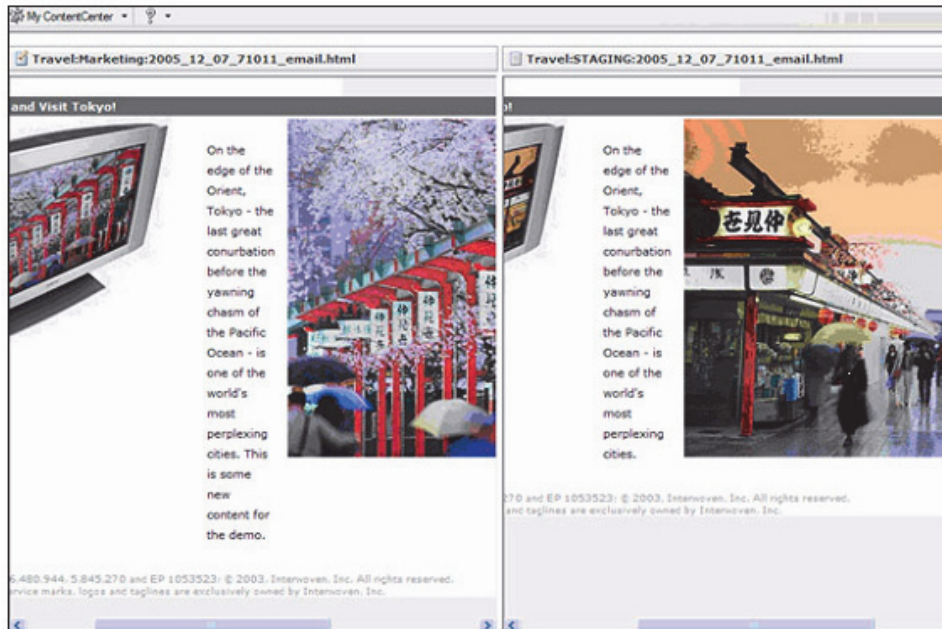


Figure 27. A visual compare of two versions of the same item.

A somewhat loftier benefit is the ability to perform date- and time-aware publishing. Digital publishing enables you to pre-schedule for content release *and* expiration, as well as any number of interim steps in between (perhaps syndication or archiving). Expiration, in particular, represents an often-overlooked benefit. You dilute the value of your current content by keeping outdated material in your repository, where (among other things) it can show up in search results.

Repository Search

Most CMS vendors offer built-in search functionality for the repository itself. (The front-end website is another story, which we'll address later.) A good search engine that indexes your CMS repository will lead to happier authors; they can more easily find that article they wanted to edit. Managers can become more powerful, because they can find the data they need to manage (e.g., specific pages and images).

At the same time, effective search integration with your content management system often requires additional, specialized resources and expertise, and you may have neither nearby. For example, you'll need to figure out whether you want to index across versions of content, and across separate physical repositories (e.g., work in progress, versus staged content). There is a case to be made for both. (An author might want to search for all instances of a single content item, regardless of status.) In some cases, CMS vendors have built-in SQL or XPath queries for this; in other cases, they assume that you will have integrated a search engine to do the job.

Remember that embedded search packages generally cannot be used in a delivery environment, and usually come with a mixture of other limitations as well, including the:

- Size of repository
- Coverage (e.g., index PDFs)

- Concurrent users or allowed environment.

Although these search packages presumably hold the benefit that they are pre-integrated, they are supplied in the hope that the customer will be motivated to purchase the fully functional version of the product from the search vendor. Check your contracts carefully.

CMS vendors that tightly bundle content production and delivery may actually employ two different search strategies: one for the CMS environment, and another for the delivery environment. The former may be a homegrown system that you may simply need to accept, warts and all. The search solution for the delivery environment may be specialized as well, but you can likely swap out with any another search engine that can spider your pages.

Content Lifecycle Services

Library Services Requirements Builder

Do you require version branching and reconciliation?

If you do, it will immediately push you to a mid-market package or better. Reconciliation can be tricky; if you anticipate doing a lot of it, take a look at the management interface of any WCM vendor that says they can handle it.

Do you anticipate needing to be able to roll back the site? How far? What elements?

You don't require a complicated system if your principal concern is being able to return to a previous version of something after making a mistake. If you need to recreate your entire site at a given date and time, you will need a more powerful WCM package. Find out what is versioned, e.g., code fragments, content elements, documents, images?

Have you already licensed an enterprise or website search package?

If you already have a standard website search package, you'll want to find out how many times your CMS vendor has integrated that package for repository search, as well as the attendant level of effort. At some point, it may be more worthwhile using the default repository search tool that ships with the CMS and avoid the hassle of integrating your own, but of course, this brings potential usability, maintenance, and cost liabilities down the road. Test carefully in either case.

Everything has a lifecycle, and content is no exception. Many CMS vendors think of lifecycle as “workflow,” but there's really much more to it, including potentially globalization, multisite management, and — ultimately — content retention and disposition.

Workflow: Editing, Revision, and Approval

Workflow is how content is modified and approved through a set of interdependent tasks that often occur in a specific sequence. A workflow process can enhance systemic stability and reliability by allowing managers to align content paths with core business and editorial rules.

Done right, workflow systems can improve adaptation time within companies of all sizes. Executed poorly, a workflow system can gum up an otherwise perfectly good publishing system. If you haven't already done so, take the time to model your existing website workflow process — what you find may surprise you.

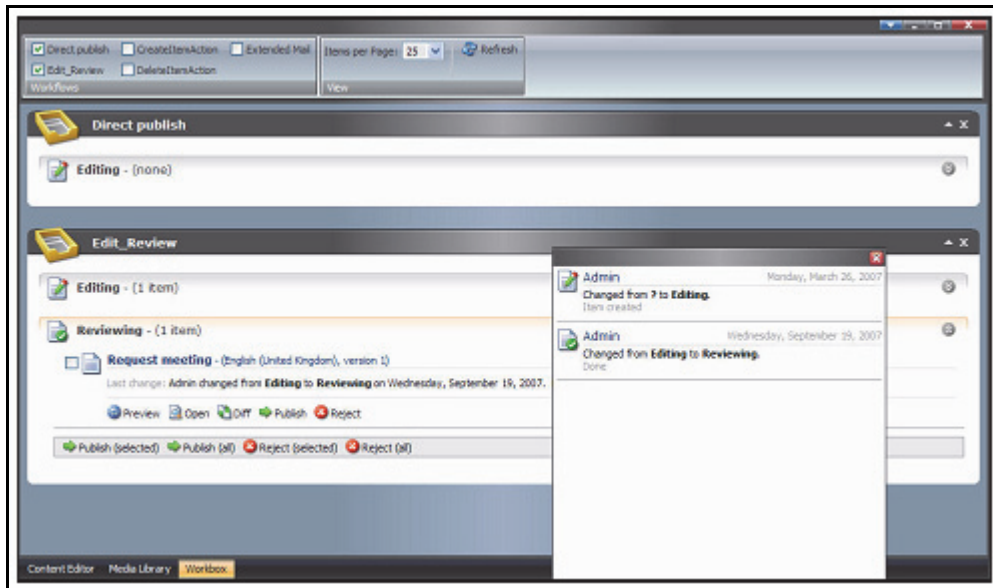


Figure 28. Workflow options are fairly sophisticated, but not always friendly in Sitecore.

In reviewing WCM packages, look for rules flexibility, including support for ad hoc workflow mechanisms. A package may contain 12 preset status identifiers, but what if one you need isn't on the list? You may well need this flexibility, because web content workflows tend to become more collaborative than other business processes that you might be automating. In fact, you should consider carefully whether most of your content *really needs* to go through a traditional, linear workflow, and whether a more collaborative environment — where content in progress resides in a kind of pooled repository — would work better for your business rules. Obviously, a big factor here is the extent to which your processes need to be audited and follow particular regulatory or legal guidelines. Just remember that extra steps mean extra work. In any case, note that you can apply different workflows to different content types. That means you need to model your processes as you model your content.

There are several different ways to model processes. We always recommend, however, starting out with different content types as the base element. Some of the best models use tabular charts, with each row representing a different content object, each column a particular state that the content resides in, with each relevant cell describing what is supposed to happen to that content in that state.

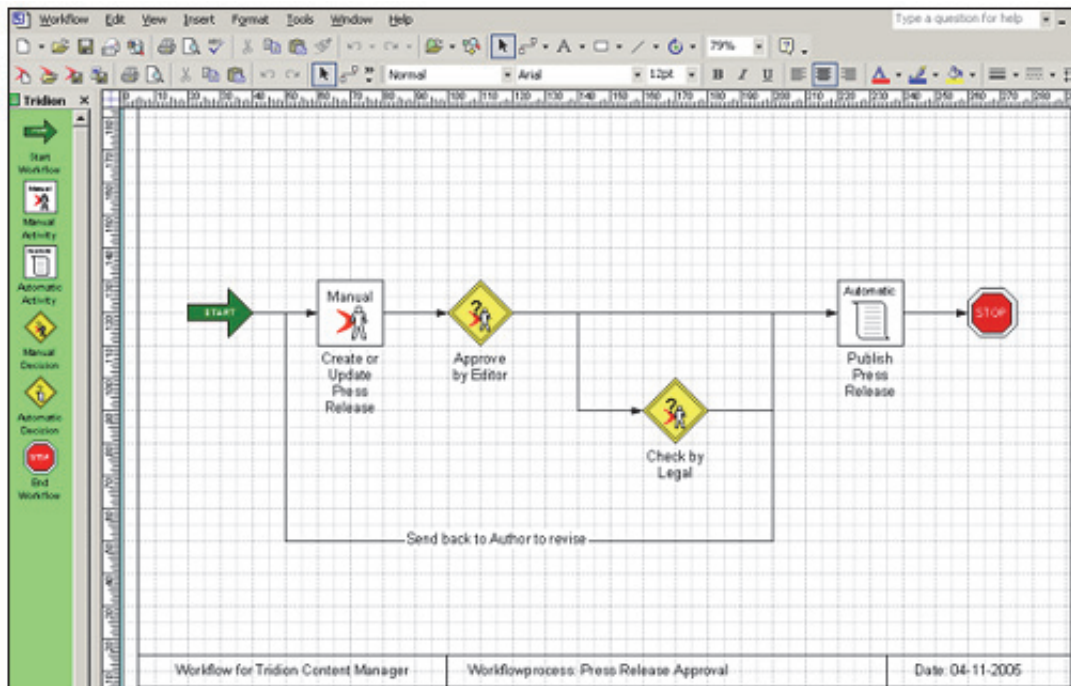


Figure 29. Modeling workflow in SDL Tridion’s Visio plugin.

Most advanced WCM products contain visual workflow modelers built in. Others — such as OpenText or SDL — will import specially filtered Visio flow diagrams (Figure 29). The keys are abstraction and flexibility. Note, for example, that in many CMS tools, content flows through different, highly generic states, in which developers and managers can apply certain specific rules. Others follow a process-flow approach, where activity and particular steps are centered on particular tasks, rather than the content itself.

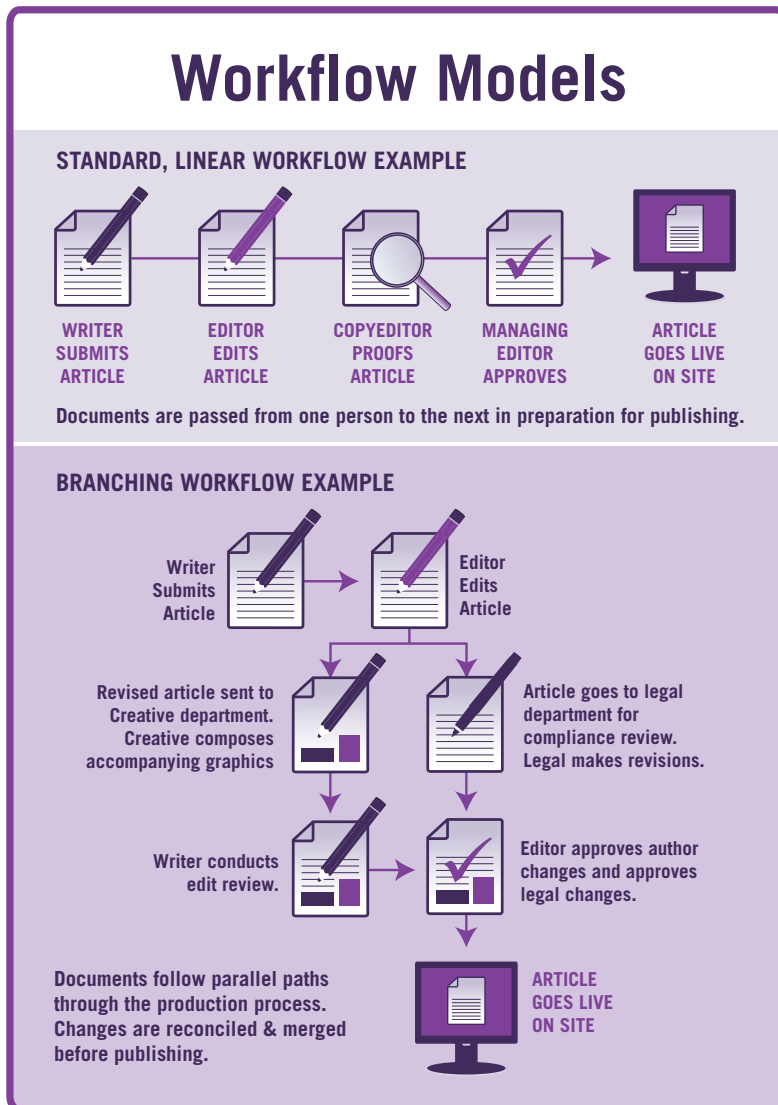


Figure 30. Linear vs. Parallel workflows.

In either case, you’ll want to model your workflow using abstract roles, rather than saying, “releases go from Bob to Lisa to Bill.” The model is actually that releases are edited and then deployed. Today, Lisa may serve as the reviewer and Bill the approver, but the people may change (or one of them may be out sick), while the workflow may not. Thinking in terms of abstract titles (Figure 30), as well as content states and transitions, will help you better understand the different situations your team is likely to encounter.

In general, however, you want to avoid the creeping process flow diagram. The simple process of modeling your workflow and introducing automation can tempt participants to insert

new and potentially unnecessary steps into the flow. Yes, business rules need to be applied and enforced; however, if you overcomplicate a process or represent a workflow in a way that doesn’t actually work in the real world, users will rebel, often by circumventing the system. Stories of this are legion. Moreover, one of the opportunities you have in implementing a new CMS is to streamline your processes — that’s where time and cost savings lie.

One final caution on workflow modeling: Make sure that your vendor can demonstrate using the same visual modeler in any CMS to make significant *changes* to an existing workflow process, as well as create new ones. If you need to use underlying scripting to make changes to a particular workflow process (which in many cases is necessary), then you can be sure that the GUI editor is out the window going forward.

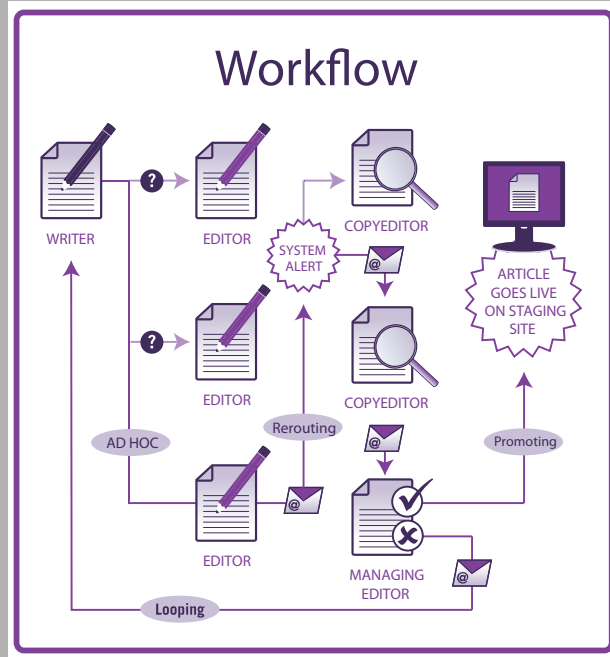
Workflow Requirements Builder

Will your workflow include branching and looping (as opposed to simple linear tasking)?

When tasks and content begin to peel off in different directions, basic workflow systems come under stress. If your workflow requires branching and reconciliation (see figure, right), or complex looping chains, make sure your CMS package can support it.

Auditing and Reporting: What do you need to know?

And when do you need to know it? Identifying bottlenecks is a real benefit. You might also want to audit individual performance and other metrics.



Unfortunately, most CMS vendors typically leave you to a report writer and your own devices. This strikes us as counter to the goal of empowering business users. Make sure you ask what is getting logged in any event.

Which is more important, speed or accuracy?

Like it or not, speed and accuracy often present a direct tradeoff. The fewer eyes that need to review a content element, the faster it goes to staging and production servers. This doesn't mean that companies with complex workflows are destined to be slow, or that firms with simple approval mechanisms will inherently publish faster. Consider the time-to-market value on each of your documents as you examine a digital workflow.

The CEO wants an item published in two minutes. Can you do it?

Your CEO needs to respond to an event in the stock markets and wants her piece online right away. It's times like these that a sclerotic workflow system can really get in the way. Make sure your system can handle unusual contingencies.

Will managers need to be able to assign tasks without any content (yet)?

Some CMS packages do not support task-based workflows, or if they do, you have to kick it off with some sort of content. That doesn't work if one of the workflows you want to have in the system is "Go create a page." Similarly, you may want to group collections of content together and have them approved concurrently. Most CMS packages can't do this.

Globalization: Supporting Multinational Web Production

In a global publishing environment, creating content and the visitors’ experience in their own native languages is critical to the multinational enterprise. There are at least two major implications for your CMS effort: This is related to — but distinct from — the earlier section on internationalization of the management interface. To support multinational or multilingual publishing, you’ll require a process for converting or adding content to create region- and language-specific pages, both with respect to the text, but also the graphics, templates, and all the other elements.

In short, globalization is about organization, technology, and systems. Perhaps predictably, corporate processes have been slower to evolve.

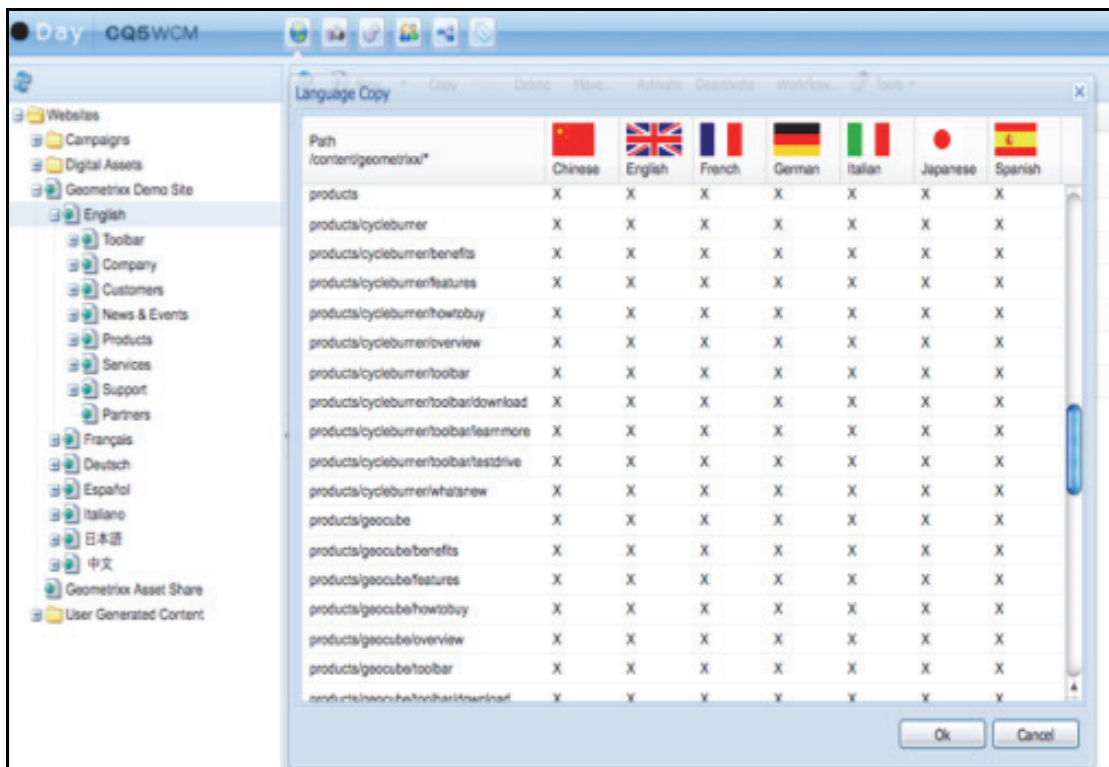


Figure 31. Multilingual, multisite management in Adobe Experience Manager.

Most global companies have figured out that they do need to localize web content. They may centralize certain messages and materials, but they leave key decisions about the way that information is communicated — and in what language — to their local offices. Indeed, localization represents far more than just text translation. Ultimately, localization is about doing business in a way that makes sense for any particular country — on or offline.

Globalization: Technology & Systems

For a multinational site working off one platform, you’ll want to consider how localization affects every layer of your CMS. Start with a vendor’s repository, specifically how it handles character sets. Look especially for UNICODE 2.0 or UTF-8 support. UNICODE is a standard that encompasses nearly 50,000 characters from all the world’s major written languages. UTF-8 is the byte-oriented encoding form of Unicode that some vendors support.

European vendors tend to have more sophisticated systems to manage multiple language versions of the same content. For example, most of the European vendors in this report have configured their version-compare systems to show the differences between a source text and the revised version, so that a translator can quickly make the same revisions in any local version.

In the meantime, a slew of professional services firms and companies with software add-ons have sprung up to help companies localize their CMS infrastructures. If globalization is important to your firm, you'll want to review the partnerships your CMS vendor has in this space carefully.

Globalization: Content, Design, & Processes

When preparing and “chunking” content for localization, the concept of a “parent” site or “parent” content (e.g., in English) and “child” sites is a key consideration. Oftentimes, templates from a parent site are leveraged for local-language presentation, and multilingual presentation should be a key consideration during the template design process. Be sure your front-end design accommodates longer languages such as German, which may cause wrapping text within navigation or other “unsightly” results. Another consideration is whether local-language content should be able to exist without a “parent” version. In some cases, organizations prefer to maintain control over local offices and websites by kicking off a localization workflow if and only if the “parent” version of the content exists.

The ability to centralize and/or link “parent” and “child” content stores, or make them independent, greatly influences your ability to maintain global consistency or enable local autonomy. In general, more object-oriented CMS tools tend to work better here, although they also typically bring more complexity as well.

There are also critical procedural issues to consider. Some CMSs, for example, will employ time zone synchronization to make sure that content is published at exactly the same local time on your different websites around the world.

Finally, major international enterprises often need to account for localization workflows. Some CMS tools interface more cleanly with globalization management systems (GMS) better than others do. A GMS can do things that aren't easily replicated in a CMS, including maintaining translation memories.

The advent of Web Services has enabled some firms to invoke workflows across applications and essentially “hand-off” tasks from system to system. In practice, this can be difficult to

engineer, and many smaller scale translation workflows rely on more copy and pasting — more so than the participants would often care to admit.

Globalization Requirements Builder

Will you need to support non-English languages or non-Latin character sets?

If so, your WCM system and any databases it uses will need to support this. The most common international character set standard is UNICODE or UTF-8. Pay particular attention to languages that read right-to-left (such as Arabic), and test how your CMS will handle these.

Is your workflow international?

Sending the same piece of content to be localized on multiple sites requires branching. Additionally, if the source content changes, you will want to trigger a revision in its localized replicas, as well. This raises a very practical issue for translators: how will they know what has changed. Displaying visual differences is critical here. For a translator needing to update the English rendition of a revised German source is vastly simplified by clearly indicating the changes to the original German version that need to be replicated.

Do your images frequently contain text?

If so, then you will need to localize those as well, and build image changes into text-localization workflows. Some CMS vendors, such as Adobe and CoreMedia, allow you to store a root image in a high-resolution format, and then publish out different renditions (presumably with text in different languages) for distribution to local websites.

Will you and your users need to search in other languages?

Don't assume that the same search engine can function effectively in different languages. You'll want to test out any bundled search engine against a repository in languages you intend to employ.

Archiving and Compliance: The Morning After

The Internet is getting old, even if many web teams act as if the web were still a frontier environment. As enterprises seek to apply tighter management to their web publishing systems (e.g., through the imposition of a CMS), attention inevitably turns to a long-term view of the web content lifecycle that ultimately incorporates archiving and disposition.

Records Management

Very often, the initial content migration into a CMS prompts at least a discussion of the reams of outdated or otherwise useless content that merits culling before the actual technical implementation. To web managers, this may present a simple act of deletion. However, in our highly regulated and oft litigious age, experienced records managers point out that managing content requires an established, editable process of identifying and categorizing actual records, and preserving and disposing of them selectively according to a particular plan.

Managing records presents a quite different challenge than simply “archiving.” Some people have the mistaken impression that records management is about hoarding everything that is

published in the course of doing business. In some highly regulated industries, it may seem that is the case. However, in most enterprises it's not only a question of making sure that what needs to be kept as a formal record is retained, but also prescribing how long it should be kept, where it should be stored, who has access to it, and when it should be destroyed (if ever).

Essential elements of a records management program therefore include a records retention policy and a set of procedures where records are classified, retention periods are defined, and destruction procedures are prescribed. Classification of records and then maintaining them with the appropriate metadata are necessary so that this information can be retrieved quickly when required.

Electronic records management software, then, operates quite differently than document management or web CMS tools. RM packages focus more on maintaining a repository of evidence that can be used to document events within an organization. While WCM repositories are generally focused on keeping as much content as possible for future reference, RM repositories are generally focused on keeping only what is necessary for a specified length of time, and then providing a mechanism for complete destruction of a record and an audit trail of the activities associated with that record. An RM system usually deploys a role-based user security model with strict filing permissions for groups of users, and very strict access controls after information has been declared a record.

Most enterprises have not tackled the formal management of web records. This partly reflects a cultural gulf between web managers and records managers, who typically operate from a different set of assumptions about managing content. Web managers tend to focus on speed, adaptability, and the immediate requirements of a particular business unit. Records managers tend to take a long-term view of information access needs, as well as a more enterprise-wide perspective on content stewardship. This naturally leads to conflict — or more typically, mutual avoidance and suspicion. Overcoming this gulf is an essential precondition to formally managing web records.

There is a technical gulf as well. Web content management systems can offer partial support to an RM plan by enabling metadata and certain lifecycle controls, but there is not a single web CMS tool on the market today that can substitute for a formal RM repository that incorporates formal file plans, disposition schedules, non-repudiation, and other staples of the records management discipline. As a practical matter, this means that any web content that constitutes a formal record ultimately needs to be locked away for processing and ultimately disposition in a separate RM repository. It also means that content items in the Web CMS repository that constitute convenience copies of authoritative documents stored elsewhere may need to be removed when the original record is processed.

A separate problem is what to do with dynamic content. Records management came of age in the era of documents, but what if web pages are assembled on the fly, perhaps with variable user input? No one has a simple answer to this. Some enterprises generate PDFs of plausible presentations. Others export all content items and templates with the hope of being able to reassemble them later (this can get tricky). Some vendors (such as OpenText and IBM) sell add-on modules that capture every visitor session and offer the possibility of recording each and every transaction.

Some Web CMS vendors are slowly beginning to grapple with the problem, but progress here has remained slow. Vendors argue privately that customers talk more about managing web

records than actually doing it. However, those enterprises with strong ambitions here should brace themselves for substantial integration work.

Archiving

To be sure, simple archiving remains important. Motivations here tend to revolve more around the needs of the content consumer than the enterprise producer. You may want to keep some older content or superseded versions of web pages available online for particular business reasons. In creating a web archive, you'll want to consider user interfaces in general and templating in particular (cuing the visitor that a newer version may exist), as well as search results (keeping archived material in a separate collection). Some vendors have explicit capabilities to smooth this process. In other cases, you may need to rig up an informal archiving mechanism, perhaps based on workflow state. That's not as ideal, since really it is a lifecycle issue — and unexpected things can happen in workflows — but it could suffice.

If this is important to you, look for systems that incorporate or at least accommodate an explicit archivist or librarian role for processing groups of content items according to filters or triggers that can be set.

Archiving and Compliance Requirements Builder

Do you have an RM program for your web content?

A formal RM program will incorporate a formal records retention schedule that defines:

- A filing classification scheme
- How long records should be kept
- Who has access to them
- Who has authority to file them
- Where they are located
- When they can be moved from active to inactive status
- When and who can dispose of them

Note that this metadata is very different from the attributes typically stored in a Web CMS. If you don't have an RM program for your web content, it behooves you to create at least the beginnings of one before archiving — or deleting — content willy-nilly.

Do you create significant original content for the web?

This is an important question for the enterprise. If the web team or some other internal group creates content solely for the web, then it is possible that some of those “originals” will constitute formal records and will need to be treated as such. On the other hand, if most or all web content — with the possible exception of certain index pages — represents content that has been repurposed wholecloth from preexisting document management systems that are under some sort of RM regimen, then the records management challenge becomes a bit simpler —but not simple.

Do important constituents need access to an archive?

Sometimes, web managers assume that constituents need access to all historical information or none at all. Usually, the answer is somewhere in between, and, as elsewhere, the solution lies in doing proper persona and scenario analysis. One big advantage of a Web CMS is that it allows you to segregate content into different types, which can have different rules for archiving. Also, you can apply different templates, and metadata, to content that has been archived, to signal consumers appropriately.

Experience Services

By “experience services,” we mean what happens when a person visits your website, mobile application, or kiosk, and starts interacting with your brand.

Publishing Services: Content Delivery

Standards Adherence: Following Web Norms with Friendly Output

“Friendly output” is a common phrase for web pages that adhere to industry standards with respect to URLs and XHTML. Vendors tend to soft-pedal this issue, but it can have a huge impact on search engine optimization (SEO), findability, and accessibility.

First, your CMS should natively generate clean, semantic URLs, like this:

`http://company.com/products/printers/printerX`

Not URLs like this:

`http://company.com/WebSite/Pages/selector.aspx?class=products&id=2345&leftnav=1`

Many WCM tools insert extra directories, as well as reams of parameters, atop perishable extensions, like “.jsp,” “.aspx,” or even “.ece” and “.cms.” This makes your pages more inscrutable, less SEO friendly, and worsens your migration pain down the road.

Some vendors counter that you can just set up your own redirects. This is a cop-out that puts the administrative burden on you. What you really need is the ability to determine programmatically what a page name will look like, along with the ability for authorized contributors to override it arbitrarily.

Now let’s turn to the actual page output. Some CMS tools auto-generate non-standard HTML, creating (among other problems) accessibility and SEO difficulties. In any CMS, you create templates, and you can usually make them as accessible or as inaccessible as you want. However, some vendors natively add extra code or constrain your options: by auto-generating tables, not supporting extended metadata for images (i.e., no alt tags), improper use of headings, and poor CSS integration. When in doubt, view the source code during demos.

Moreover, remember that your CMS also empowers non-developers to add content, and the material they publish may not be standards-based or accessible. The ubiquitous, rich-text editors that accompany most Web CMS tools can be particularly problematic here, especially if you don’t (or can’t) lock them down properly. Some vendors have modified them to do things like require table headings and alt tags for images. Other vendors offer optional modules for validating contributed text against different accessibility standards.

Finally, some CMS platforms — particularly those that rely heavily on discrete “modules,” may assume a stylesheet model of heavily in-line CSS — you likely don’t want that. Ensure that any system can work with your CSS framework.

Multichannel & Syndication Services: Publishing Alternate Formats

Whereas we addressed “aggregation” of disparate content in the production phase of the CMS lifecycle, you could consider this feature *dis*-aggregation — breaking apart your web content for different delivery mediums beyond the web browser. These include mobile devices, email, print, kiosks, and syndication streams.

The most important consideration with respect to publishing to other formats is that technology may likely prove to be the least of your challenges. Techniques for exporting content to mobile formats, email campaigns, or print layouts is well understood and fairly transparent, especially if your content resides in XML. The more difficult problems revolve around developing and maintaining a content model that will support repurposing information

in logical chunks, and still be able to evolve flexible processes for dealing with different media. Let's take a look at print as an example of the broader issues.

Print

Integrating print publishing and a CMS is complicated. Native web content tends most often to find its way back into print via PDF files, perhaps because the tools to do this are fairly mature. In contrast, converting from XML to word-processing or desktop publishing files can still be a clunky process. One solution is to address this by completing the “single-source” loop with a unified authoring environment — almost surely XML-based — that will publish successfully to all your digital and print venues. This is tougher than it sounds, partly because the print world is not completely XML-enabled (let alone completely digitized), and partly because it is often very difficult to “round-trip” back changes that are made in the pre-press environment in tools like Adobe InDesign back into the original repository, to maintain that sole source of the truth. In short, there is no right way to go from web to print; it depends on which alternative is most efficient for your people and processes.

If you are interested in single source, multichannel publishing (and are willing to trade off potentially significant WCM capabilities), you will likely need to investigate an entirely different class of (very niche) vendors that develop “Component Content Management” technology. Some vendors offer varying degrees of Adobe InDesign integration, but test them carefully; this is an inherently tricky process.

Syndication

The case for syndication is simple: You can obtain more value from content by making it accessible in more places. However, think quality in addition to quantity when considering the advantages of additional venues for your content. For example, if most consumers of your product enjoy close relationships with your distributors, then you can make everyone's lives easier by syndicating content to your distributors' websites, where your materials can be meshed with theirs in a more compelling package.

This example prompts us to address the difference between traditional media syndication and B2B syndication. Many people think of syndication solely in terms of media or publishing companies — that is, when content *is* product. Syndication may be even more powerful when content serves as *collateral* to the product or service. If content is a key differentiator for your firm, then the ability to get it out to prospects and customers wherever they may be found could be critical to maintaining your unique niche and identity in the online marketplace.

The actual mechanics of content syndication are fairly easy, especially if you already store content in XML or have the means to transform your data into that format. Syndication standards remain somewhat gelatinous right now, but all of them — including the commonly used RSS, ATOM, and ICE — are XML based.

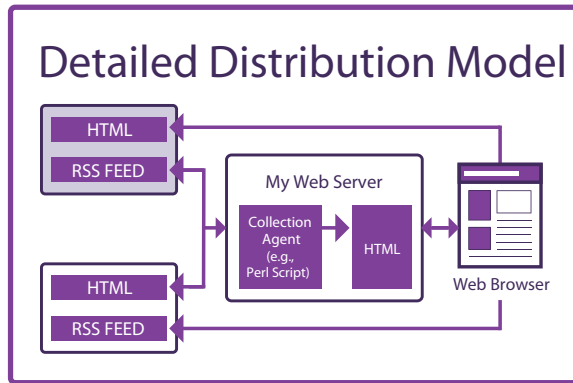


Figure 32. RSS feeds XML, and then converts to HTML headlines.

Like most syndication formats, with RSS or ATOM, you store the article on your web server, where it can be retrieved, parsed, and transformed into suitable HTML by subscribing sites.

While this approach is elegant, many potential subscribers will not know what to do with an XML file. For those, you will need some sort of uniform HTML-rendered feeds that they can pop into their sites.

Many WCM systems provide some basic syndication tools, if only the means to generate XML and HTML files for subscribers. Not all tools, however, will generate

RSS or ATOM feeds automatically, and you may need to write some custom scripts to do so. On the other hand, other CMS systems can output thematic feeds based on metadata, either as static files or a dynamic feed. In the latter case, stay mindful of performance, since feeds can be tapped by aggregators or feed-reading clients even more frequently than visitors hitting your HTML pages.

Multichannel & Syndication Requirements Builder

Will you be generating print materials from web-originated content? If so, you'll probably want an XML version of your web documents that you can more easily import into desktop publishing systems. First ask yourself: Could the printout from a browser version of a printer-friendly page suffice? For some purposes, it might.

Will you be sharing content with partners? If so, syndication standards will make everyone's lives easier. Even within closed ecosystems, syndication is an efficient and simple way to share content. In the absence of true inter-enterprise WCM functionality among the vendors reviewed in the report, syndication may be the *only* stable way to manage content collaboratively.

Mobile: It's about Device and Content Adaptation

Most WCM vendors will tell you that they support mobile delivery. However (as with any other feature), what the CMS can do and how it actually does do it can be two very different things.

The extent to which vendors support mobile delivery varies substantially in both form and depth. You need to be cognizant of several key issues when selecting a product and creating a site for delivering content to mobile devices.

Remember that generating content from a database to a mobile device can be easy to execute technically. The trick comes in content organization and the user experience. While you're evaluating capabilities in WCXM systems to provide a first-class, mobile experience, consider how you can do the following:

- Separate content from presentation
- Adapt for device-specific capabilities and limitations
- Adapt content to a specific device
- Adapt content to a specific geolocation
- Adapt templates for mobile
- Adapt media assets with proper formats
- Adapt content production for mobile contributors
- Incorporate HTML5 capabilities



Figure 33. Device-specific preview may depend on what mobile emulators are built into your CMS. Note that each target will likely require a separate template or template variant, unless you implement HTML5-flavored responsive design.

Many vendors today will claim to support “Responsive Design” (RD). Responsive Design enables you to create a single template and set of stylesheets that will allow for graceful content re-ordering against different mobile environments. In general, this is an empty boast; RD is really about the templates and CSS you create, and not really a native attribute of the CMS, itself.

Separating Content from Presentation: Pages vs. Components

In theory, any WCM system should be able to deliver content to a mobile device. The basic tenet of content management is that it separates raw content from its presentation — using a presentation template applied on raw content to transform it into a templated web page. Publishing to mobile devices then becomes a matter of creating yet another template to format this content to a small screen. You typically write a script that identifies the visitor's browser, and if it is a mobile browser, your CMS applies a mobile template and streams out content. To take this a step further, there are specific libraries for device recognition that allow you to

further tailor content based on certain devices and models that you're delivering to, and we discuss this topic in greater detail later in this section.

Note, however, that many WCXM platforms don't actually maintain a strict separation of content and presentation. Creating pages via in-context editing and heavy use of rich text editors (RTEs) is convenient for contributors, but makes it difficult to repurpose content in other formats. Hence (at the very minimum), your CMS should be component oriented and not purely page oriented, so that you are able to store raw content components with minimal markup information. In the product evaluations, we identify when a solution is more component, versus page oriented.

Perhaps the larger challenge here is one of information architecture. Content organization and the resulting user experience on a mobile device is quite different from that of a PC for a variety of reasons such as:

- Mobile devices usually have limited screen sizes
- Your standard “pages” may need to be broken across multiple screens, and each screen must be short enough for a quick read
- Your navigation may need to be shortened
- You may need to create mobile-specific versions of some content that are more truncated than your web content
- Entering text on a phone keyboard is tedious; users should be able to select options from menus, rather than typing long strings of text
- All buttons should be short, concise words, clearly indicating what will happen when they are pressed

Thus, it becomes very important for you to segment your content carefully into mobile-friendly chunks and to consider the different information architectures required in each media format. The technical transformation from XML may be trivial; the planning beforehand is not.

In the face of this potpourri of device profiles and capabilities, you can choose among three broad choices, each increasingly rich — and more complicated.

1. Disregard all of the variation and use a simple “light” template that creates a reasonable display for (almost) any small screen
2. Create separate templates for major types of devices (which some vendors do when they provide support for major smartphones)
3. Vary how you present the information based on screen size and multimedia support for every individual handset using mobile device recognition libraries for those who don't have iPhones, Androids, or other smartphones, but still have browsers on their phones

Adapting Content to a Specific Device

Once you identify a particular device's capabilities and profile, you may want to adapt the actual substance of your content based on those capabilities. For example, you might want to show completely different content to Android users than to iPhone users, because your

research shows different demographics for these two handsets. Or, you might want to show a special campaign to Nokia users that you don't want to show to Sony Ericsson users.

Adapting Templates for Mobile

Device and content adaptation implies that you can deliver the same content or variants on the same content, via different design and layout templates. This requires a one-to-many relationship of content to templates. In some lower-end WCXM systems, templates are bound tightly to content types, and it is not easy (or even possible) to deliver the same content through many different filters. Your templating subsystem should also provide the ability to include placeholders and populate them based on the rules or policies that you have defined.

Adapting Media Assets with Proper Formats

Based on the kind of devices you want to support, you need the ability to encode images, audio, and video files, depending on how they handle media files. You probably will need to transcode (convert from one encoding to another) these assets to many different formats. You may also need to “gracefully degrade” from one format to another (e.g., from Flash to JavaScript + HTML), depending on device support. Some CMSs can pre-generate and store these assets in multiple encodings or convert them on the fly. In any case, you need to know whether any technology manages only one “master” format and creates others on the fly, or pre-generates all formats and manages them collectively.

Adapting Content Production for Mobile Contributors

Most of the above considerations are for delivering content, or for “plumbing” features that enable you to deliver content to such devices. You'll also need to provide your authors and content-entry people with features that enable them to use these devices to submit content. Ultimately, the tool should provide the ability to upload news items, images, and other digital assets, participate in workflows for publishing, and more, using a handset. If you need your content editors to use mobile to contribute content, think carefully about using rich text and in-context entry mechanisms. As previously noted, while usage of RTEs and in-context editing features might be beneficial for an occasional business contributor, they are not good tools if you want to be able to repurpose and reuse content. Thus, make sure that the WCM system provides simple, forms-based content entry mechanisms to enter content. Additionally, find out if the tool allows you to create multi-page forms so you can split your forms across multiple screens for content entry using a mobile device.

HTML5 Brings New Mobile Browser Capabilities

HTML5 is the next version of the web's de-facto HTML standard. Many evangelists would have you believe that HTML5 is a panacea and will obviate the need for downloadable applications. To be fair, HTML5 has some relevant capabilities for mobile development:

- It allows you to create an experience that is closer to what is offered by native applications in terms of look & feel, ease of use, and performance
- It will have capabilities for offline storage and geolocation

- It does the above in a standards-compliant way and presumably allows you to have a much more scalable and efficient model in terms of the types of devices that are supported by your applications

However, enterprises face an equal number of challenges with HTML5:

- Despite all of the hype, HTML5 is still under development; although many devices support it, there's much more to be done before it's ready for prime time
- If you have applications that need to take advantage of device-specific capabilities, HTML5 will not suffice
- In production, HTML5 relies on a bevy of adjacent technologies, such as CSS3 and JavaScript libraries; thus, creating a rich HTML5 application could require almost as many (sometimes even more) resources than you would need to develop a device-specific app

Digital Marketing Services: Engagement, Measurement, and Optimization

Whether you are an intranet manager, digital marketer, or leading up a public website, you are likely publishing information because you want your visitors to do something. You are no longer just managing content, but managing your visitors' experiences and their engagement with your brand, site, product, and campaign. Modern CMS products should be able to provide support in your engagement strategy execution across all delivery channels — web, mobile, print, and local.

Long gone are the days of flat brochure sites, as the web as we knew it has evolved beyond Web 2.0 into a two-way street of conversation and engagement. As web visitors become more pressed for time (and more impatient), marketers have to make the best out of every opportunity, engage them, and turn them from prospects into customers. This is where additional technology capabilities can complement a CMS to deliver more to the increasingly sophisticated expectations of consumers.

The terms “Digital Engagement Management” and “Web Experience Management” (WEM) are not new. WCM vendors like Oracle, OpenText, and HP have been toying with this notion for years, but broader interest in experience management hasn't arisen until recently, presenting a natural evolution from “personalization” that we've seen since late 1990s.

Marketing hype aside, WEM should not be mistaken for a set of new tools. We define Web Content and Experience Management as a practice and strategy on how to engage with your audience, using specific engagement and digital marketing tools that may (or may not) be available as part of a WCM product.

A CXM strategy may include a variety of constituent pieces, including content targeting, delivering the right content at the right moment to the right audience (with some automation), measuring the response, analyzing and optimizing content, socially collaborating and conversationally engaging, multichannel delivery, social/CRM, and other systems integrations.

While traditional Web CMS vendors are heavily touting their WCM services today, it is premature to call out WCM as a distinct product category, separating it from the web content management market.

Additionally, vendor claims of WCM capabilities should not distract you from your seemingly pedestrian — yet still difficult-to-achieve — goals for managing content itself. Finally, you need to assess whether it makes the most sense to obtain such specialized services from your CMS vendor, or from third parties.

Personalization & Segmentation

We define Segmentation as a practice of identifying various segments of online audiences that are marked with distinct interests, online browsing behaviors, or socio-demographic attributes. Segmentation allows marketers to cater more efficiently to those different subsets of prospects with targeted content, and to improve conversion.

Personalization is about generating custom content or presentation for individual visitors. Since it fundamentally concerns itself with mixing and matching content elements — and seems on the surface to be a good idea — many CMS tools have wandered into the personalization space. This is because some use cases (sales-oriented sites and some intranets) make a decent case for personalization, and content managers understandably often want to manage this sort of interaction within their regular editorial interface, rather than within a separate Portal or personalization tool.

There are several elements to Personalization & Segmentation, including:

- Visitor behavior
- Explicit (active) and implicit (passive) personalization
- Profiling and content targeting

Depending on the level of engagement, you can address groups of various sizes, as well as single individual users that you want to target with appropriate content. Segmentation focuses on groups of users, whereas personalization focuses on individuals. Profiles containing predefined characteristics can be assigned to either individuals or groups. Determining the way users behave on your website (scrolling, clicking, and navigational patterns) further tailors the personalization process.

You should look at personalization from many multidimensional perspectives — who your audiences are, how they behave, where they are located or come from, and what kinds of additional data you can gather about them. In the case of explicit personalization, you go by the data supplied by users voluntarily (e.g., they fill out a form and indicate age, industry affiliation, socio-economic status, etc.). Thus, your site visitors are actively supplying data for a more personalized experience. The downside for you is that users choose which content they want, and you don't have much influence on the process aside from comparatively simple judgments.

Here's a summary of several approaches to personalization:

- **Profile based:** In this model, site owners and editors create custom pages or content sets that are served to predefined profiles of users. Profiles can be attached to individual users, groups of users, or both. This approach well may be the easiest to automate.
- **Rules based:** In rules-based personalization, some level of artificial intelligence is employed. Site owners and editors establish business rules and conditions that, if met, affect the display of content on pages within a site. For example: An online gourmet

store is overstocked in French Roast coffee; the site software recognizes this in the inventory database and automatically places promotions to users who have bought coffee before.

- **Preferences based:** In this model, users indicate their preferences and options, usually during an initial session, so that dynamic pages can be delivered on subsequent visits to the sites that better match their interests. Users are typically assigned a user name and password for authentication; cookies are often also enabled so users aren't required to log in each time they revisit the site.
- **Recommendation and community based:** In this approach, site software uses complex statistical techniques and automated analysis to determine elements of a personalized page. An example of this is Amazon's "people who bought this also bought that" service. This approach can also incorporate inference technology: Learning users' behavior by monitoring where they go on the site and/or what they search. An advantage of this approach is that users don't have to do any work to see a personalized view of the site.
- **Adaptive:** This type of personalization usually extends beyond rules-based targeting that is based on certain criteria, such as search terms, geolocation, or point of entry. Adaptive personalization employs more subtle properties, such as intent, mood, and online behavior in combination with data mining and artificial intelligence algorithms as main drivers behind delivering relevant content. This is one of the more complex and multi-dimensional hybrid personalization approaches. Performance requirements should be analyzed especially carefully, as there's a potential for a rather sizable architectural footprint in this case.

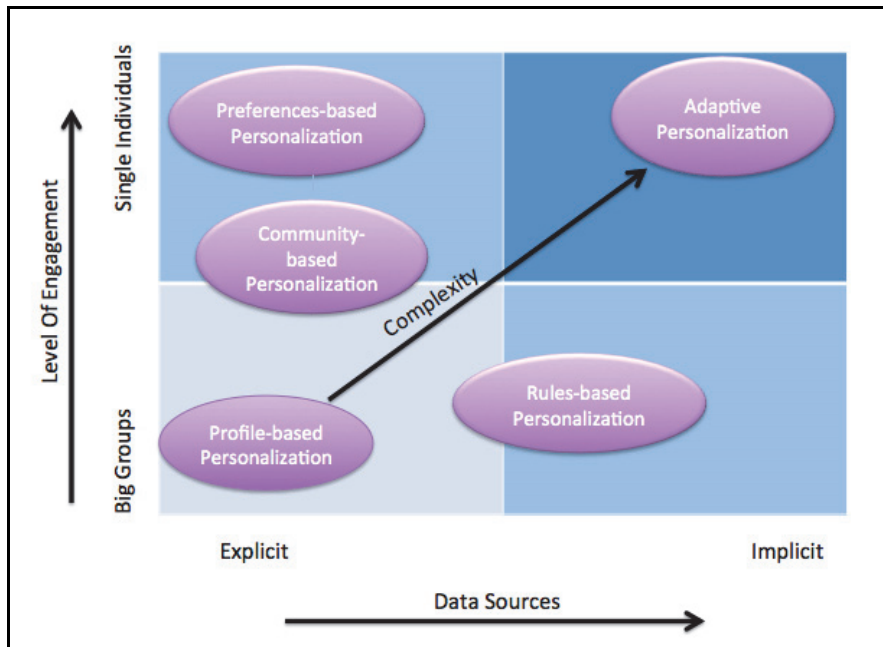


Figure 34. Multidimensional attributes to personalization may increase implementation efforts, regardless of the feature set provided by a CMS.

The development of a personalization strategy goes beyond designing the database and code to feed dynamic, personalized content to users. In many cases, you will need to make sure a Web CMS has an interface for audience management to enter custom user information, manage groups and profiles, or to override automatic processes. You will probably want user- and group-based reporting that typical web log analysis tools will not be able to provide. The effort to implement and customize these administrative and reporting tools should not be underestimated.

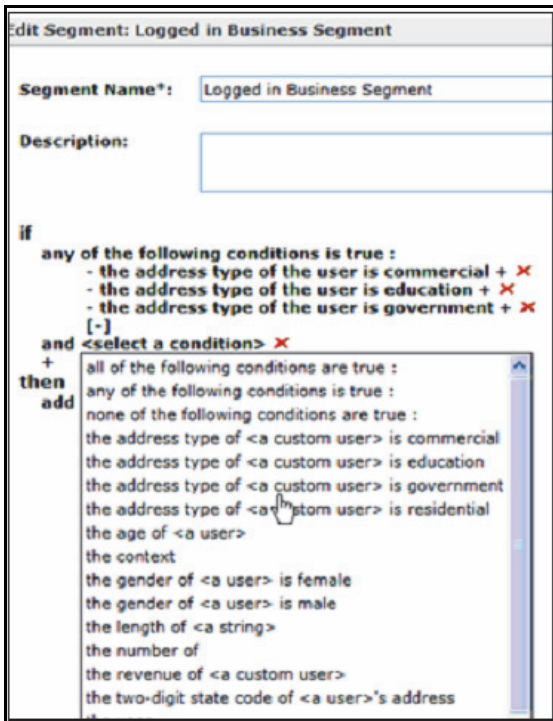


Figure 35. Creating rules may not be as simple as it looks.

Depending on how your profiling is set up and how fine-grained it is, bigger groups might be easier to manage, but the depth of personalization may suffer, since targeting specific content on individual user levels can be very unpredictable. While a Web CMS can help you by lending a set of features to drive your personalization efforts, a bigger effort lies in the design of your strategy, and picking an efficient and feasible approach.

As with all other attributes, be sure to get input from all responsible parties, including editorial and technical staff before scoping out personalization requirements.

Challenges to Personalization

With implicit personalization, you are looking at often more ephemeral data points. Over time, this implicit data can get persisted and become part of an explicit profile. Implicit and explicit often go hand

in hand, building on the data source and nature of the personalization logic. This requires applying more complex logic.

Investing only explicit or only implicit personalization might feel like an easier undertaking, but it will hardly yield many valuable results. Most often, organizations employ a clever mix of many (if not all) types of personalization.

Segmentation and personalization have proven to be a highly specialized and quite complex domain. You should not assume that your CMS vendor’s personalization capabilities will meet your needs without thoroughly testing it first. Personalization can be very resource intensive from a design and publishing perspective. It takes a lot of effort to define and implement business rules against various use cases and then test them carefully across various permutations. Dynamically serving up custom pages can add dramatically more server cycles to what may already be a very dynamic publishing process. Beware of slower page loads.

When employing complex algorithms, doing segmentation can be very hard, as it is a lengthy endeavor to get statistically relevant data, even for smaller segments. In short, the hidden costs

of segmentation and personalization can easily equal or exceed the initial licensing and development costs.

Privileges Management: Controlling Who Gets to See What

Whether you manage an intranet, extranet, or public site, you may want to restrict access to certain content areas to authorized external visitors, e.g., your customers, subscribers, or members. Some CMS tools have built-in mechanisms for segmenting and authenticating external users. Many managers perceive this as personalization, and indeed, it is a kind of reverse segmentations.

Many CMS packages treat this as an extension of roles that internal actors play in the system, but it is considerably different. Unless they are actually contributing content themselves, consumers of information are likely to interact with your site very differently than internal authors and editors. For example, external visitors have special usability needs with respect to authenticating, obtaining lost passwords, and understanding explicitly what areas may be off limits to them (and why).

Traditional access control lists for authenticating external users don't always scale very well. As with internal Role Management, many companies have found that LDAP provides a useful, universal infrastructure for managing access for partners and preferred visitors. Those firms will want to seek out LDAP-compliant CMS tools.

You can use the default authentication system built into your web server (such as working with “.htaccess” files in Apache) to control logins. This presents a very simple and convenient approach, but often means working with crude interfaces, and that can bring management headaches to large numbers of users. Webserver authentication systems typically do not offer more advanced security features, like preventing simultaneous logins with the same credentials.

Personalization and Segmentation Requirements Builder

Is there a pressing user-focused need that personalization could solve?

You should sense a palpable market need before proceeding with a complex personalization effort — you shouldn't just do it because you can or because your application server or WCM software supports it. If your answers are all focused on internal needs, then your investment may not provide a good return.

Do you require segmentation or true personalization?

Segmentation focuses on affinity groups, while personalization works at the individual user level. In many instances, limited customization can save time and money in developing a targeted information system. It is also a good way station on the path to more advanced personalization.

What is the ROI you are expecting from a personalization effort?

Would your online sales benefit from a one-to-one marketing approach? Would you be able to increase customer satisfaction? Note that there are important reporting and privacy management issues to consider that may add overhead.

Personalization and Segmentation Requirements Builder (Continued)

What personalization models will you employ?

As noted above, there are varieties of models, and they each have structural consequences. Do you have the database infrastructure to support a filtering or rules-based approach? More critically, which model or models do your target WCM packages support? A package that offers “personalization,” but strictly on a preferences basis, cannot be expected to filter your visitors collaboratively.

Do you need a separate personalization package?

Major products are available from specialized vendors. However, they may not be pure personalization tools, and have dispersed focus on campaign management, CRM, and the e-commerce market.

How granular will the permissions need to be?

You may have only one set of external users to authenticate, all with the same permissions. Alternatively, you may need to account for multiple levels of users to multiple private content areas, or different views of the same pages and content types depending on the user. Make sure that your privileges system meshes with your approach to content management. If the latter entails burning static pages for publishing to a web server, then you will need page- or directory-based permissions.

Site & Campaign Analytics: Applying Metrics to Content and Experience Management

“How are we doing?” Managers ask this question of web teams as much as any other business unit does. One way to measure how your website is performing is through formal site analytics packages (e.g., Webtrends, Adobe Omniture, Google Analytics, etc.). These tools can usually go beyond measuring simple traffic to include analysis of transactions, behavior, and referrals at a much deeper level. Still, they sometimes have difficulty with dynamic content and accurately reporting on CMS-generated sites, which often requires some custom integration.

The past few years have seen a growing awareness that content managers need ready access to analytics. Ideally, an editor working on revising a content item would know its Google page rank, how frequently and when the item was accessed, and whether and how it led to any transactions. In short: How effective was the content?

Many CMS vendors say that they “integrate” with analytics packages, but beware: More often than not, it means they can enable you to drop the appropriate JavaScript analytics tags into your templates as a discrete element, perhaps assigning some specific values to particular parameters at publish time. That’s relatively trivial stuff most of the time, but still usually requires some customization by a developer (Figure 36).

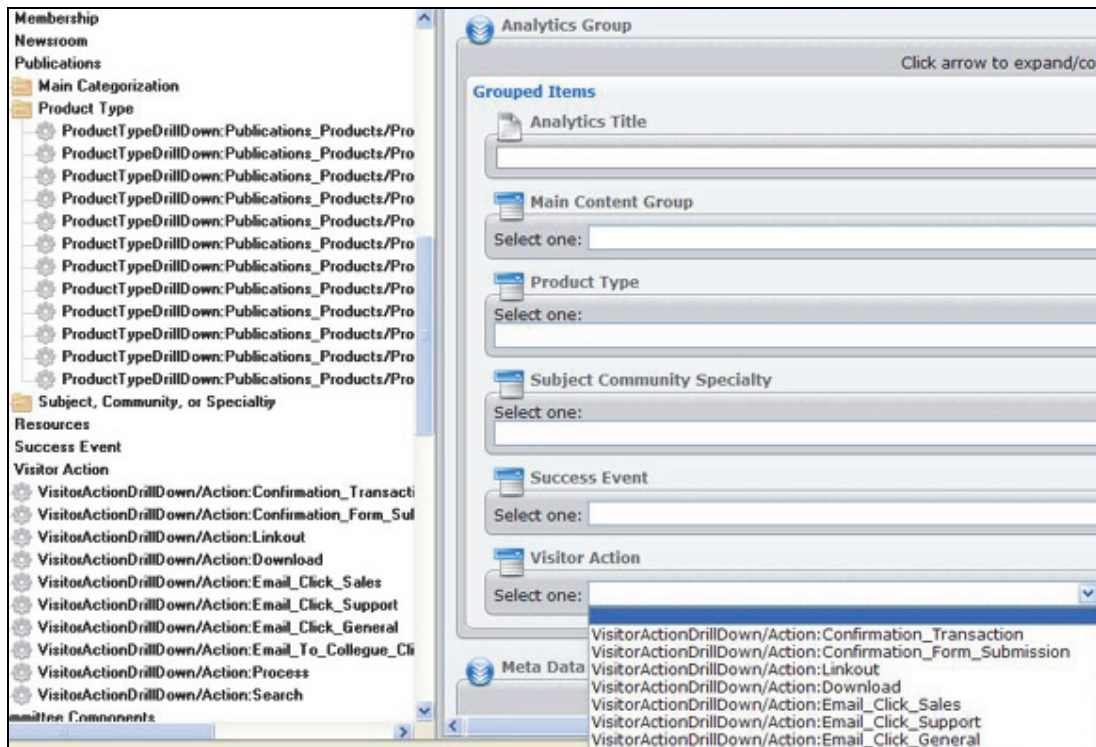


Figure 36. Under a special customization of Ingeniux CMS, a web analytics manager can use the editorial interface to configure the Webtrends page tags for a particular category of content.

Perhaps the bigger prize here is incorporating reports right into the editorial interface so that editors can act on performance data.

Some CMS vendors have responded by incorporating their own homegrown analytics packages that integrate nicely into the editorial interface and are natively aware of important metadata, such as page title, last modified date, author, and so forth. The downside to this approach (not surprisingly), is that these tools tend to be much less robust and less transferable than third-party packages. As an alternative, some CMS vendors have partnered with a third-party analytics supplier to perform advance API integration to expose data in the editorial interface. This can work well if you prefer that tool, but is a wasted effort if you prefer an alternative that the vendor doesn't support.

Ideally, your CMS vendor would allow you to plug in your choice of analytics package into their managerial interface, overlaying those reports with information from within the CMS. Few vendors have accomplished that yet.

Testing & Optimization: Could You Do Better?

Another question frequently asked by web managers and digital marketers is “How can we do better?” Marketers have always wanted to test different approaches. Why should the web be any different? There are two broad kinds of tests:

1. A/B Testing, where you measure how two different alternatives perform

2. Multivariate Testing (“MVT”), where you measure how multiple variants to a page or component perform

Of course, there are many different ways for a page or page component to “perform,” and as such, the sophistication of your tests may vary. In general, though, most testing services operate by having you swap out a page component with some JavaScript or iframe element that makes a call to the testing service or functionality. The test module then delivers HTML based on whatever test format. For example, if it’s an A/B test, every other page hit will deliver the B variant rather than A. It can get tricky from there though, since the service will need to do things like track visitors over subsequent sessions to make sure they don’t see a different variant the next time they visit.

Analytics & Testing Requirements Builder

Do you already have a traffic analysis package or service?

If so, ask your vendors about the level of effort to integrate it into the admin interface, ideally on a content item-by-item basis. At a minimum, you’ll want to know their experience with inserting the relevant code (typically JavaScript) into templates and picking up CMS-specific data (such as page title). Be wary of analytics packages that insert session variables into URLs.

What core stats do you want to expose to managers?

Omniscience is wonderful, but how much data is good enough? If basic traffic statistics will suffice for your content managers, then perhaps the simpler reporting module that bundles with your CMS will work well for you (and certainly ease the integration burden). Marketing or merchandising managers could still presumably use a separate third-party package for their own needs, but note that two different package may show different results on the same items.

Campaign Management

Digital marketers today seek to deliver and manage campaigns among multiple audiences across multiple channels. Of course, promotional campaigns come in many flavors, and as elsewhere in this discussion, you have a fundamental choice:

1. Choose from among a wide variety of standalone campaign management products
2. Adopt the marketing campaign management functionalities that are increasingly extant within Web CMS tools — albeit with varying degrees of depth and extensibility

Promotional campaign services could include the following scenarios:

- Utilize personalization capabilities with existing segmentation rules to deliver campaign-based content targeting
- Email campaigns for segmented newsletters and promotional mailings
- Mobile targeting that can be based on device, location, and other attributes
- Campaign landing page management with A/B and Multivariate testing, lead generation and other web forms, and campaign microsites

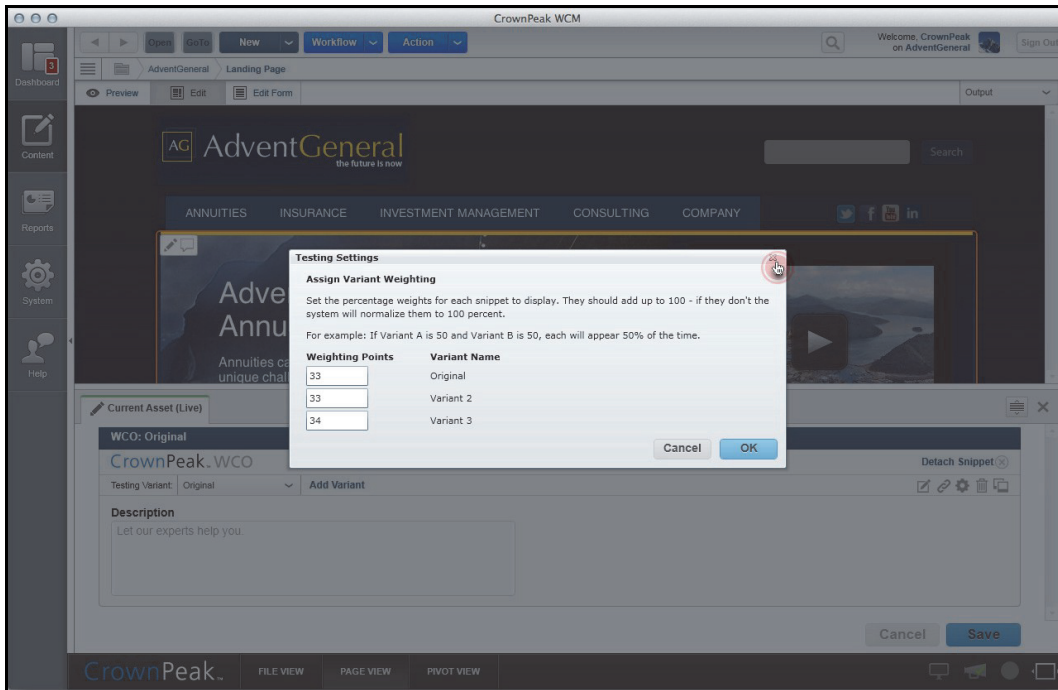


Figure 37. Assigning testing values in CrownPeak CMS.

Intermingling with web experience management and personalization, campaign management services usually fall into some sort of an “online marketing suite” variation. Sometimes these capabilities get built into the CMS base, and sometimes they are treated by CMS vendors as separate add-on modules, priced accordingly. You should very carefully research all of the different options and capabilities of promotional campaign management tools offered with a certain CMS, as there are vast differences in this field.

For example, some CMSs offer useful shortcuts for selecting and formatting content for email campaigns. However, if you intend to do mass mailings, we recommend finding specialized software or hosted services designed specially to do that. Sending mass marketing email campaigns can be hard on your servers, your network, and other IT systems. Not surprisingly, the hosted CMS vendors (e.g., CrownPeak and Upland Software) offer comparatively advanced capabilities for integrating CMS repositories and mass email campaigns.

Social Media Integration

Your organization can participate in public social media by:

- Pushing content to social networks
- Ingesting and aggregating social content, activity streams, or otherwise exploiting individuals’ social graphs
- Social media monitoring and analysis

Let’s start with the first. A recent trend in web content management is publishing to social networks — Twitter, Facebook, LinkedIn, et al. — out of your Web CMS. Why would you use a CMS tool designed for organizational publishing when you can employ lightweight client

applications for interacting with social networks as an individual? There are at least five reasons why you may want social media publishing functionality in your Web CMS.

1. **Approvals.** In theory, your CMS can allow you to put updates through some sort of workflow. In practice, you'd probably never do this for individual tweets or status updates, but you might want to have someone sign off on corporate-account tweets or updating a corporate Facebook page. This may be a weaker rationale, but vendors say some customers are doing this.
2. **Scheduled publishing.** This is where CMS tools typically excel: time-based publishing. Of course, there are other tools in the Twitterverse to schedule tweets, but they don't all work well, and by scheduling updates through your CMS, you can time them more arbitrarily to coincide with content you may be highlighting. Ditto for Facebook pages.
3. **Alerting.** Here you're basically replacing third-party tools (that convert website RSS feeds into tweets or status updates) with an internal service that's potentially timelier as well. This could prove to be especially handy for broadcasting via internal micro-blogging networks.
4. **Archiving.** If you want to keep all your social content and messaging, do you trust the services themselves to maintain them long enough for your needs? This is a particularly big deal in regulated or heavily litigious sectors. Like it or not, e-discovery requirements are real, but they don't have to keep enterprises from participating in social networks.
5. **Repurposing.** A CMS can allow you to repurpose tweets, status messages, and the like to a separate collection, or push them elsewhere without having to extract them from that social network via its API. This is also related to archiving: if Twitter somehow goes away, or your LinkedIn account gets zapped, your content stays accessible (and searchable) in another location of your choosing.

Just remember that publishing to social networks is not the most important attribute of a CMS. It's often something you can jerry-rig yourself, or you can employ various third-party tools. Also, don't confuse publishing with interacting. Your CMS won't help you participate in a LinkedIn discussion or retweet an interesting post.

Now on to aggregation. Ingesting and aggregating social content can happen in a multitude of forms with public activity streams, Twitter feeds, status updates, and social graphs being pulled onto your web pages. Few CMS tools (mostly those with blogging origins like WordPress) have native services for this. In some cases, it's available as a third-party module.

In terms of social media monitoring and analysis, you are most likely looking at integrations with other systems that have been specifically designed for the purpose of listening to and measuring social media conversations happening around your web content. Few (if any) CMS tools will offer this functionality natively. However, some vendors are starting to incorporate social media monitoring via integrations with tools. Be aware that there are multiple challenges in this arena, including effective monitoring and sentiment analysis. Some search vendors are finding renewed interest in this area, trying to tackle the challenges of mining and indexing social content within the enterprise.

Community and User-Generated Content (UGC)

Unlike social media integration, community tools refer to functionality that happens on your site (although this does not necessarily mean on-premise), and are more in your control — from design to execution.

One word of caution here: the majority of organizations as well as social community and collaboration and CMS vendors are still struggling in this area. One of the challenges (albeit there are attempts to address it) is community management and content moderation. In both of these areas, you should plan accordingly not only for a technology investment, but also for HR: many organizations have instituted new roles of “community managers,” and moderation is done partially by machines, but not without an involvement of a human eye.

Since the very first “CGI”-based web applications, visitors have been able to contribute content to a website, but now we have a fancy name for it: User-Generated Content, or “UGC.” What’s universal now is the ethos of participation, where visitors expect to engage with site owners rather than just passively read information. UGC-based applications run the gamut from basic utility applications to full-blown community services:

- Polls / voting
- Surveys
- Forums and chat
- Wikis
- Visitor-commenting facilities
- Sharing

Admittedly, all forms of social content require some participation from the users and the community. However, social tools are usually defined in terms of:

- Tagging
- URL shortening
- Ratings and ranking
- Blogs
- Microblogging
- Status updates
- Social calendaring
- Social collaboration
- Location-based services (e.g., Foursquare)

Content management packages that provide these UGC and social applications natively can usually boast integrated metadata and (where needed) workflow frameworks, as well as ease of applying a common look and feel. Perhaps more importantly, they can allow content managers to control other types of information exchange without resorting to separate administrative interfaces — and a separate technical team to make it all happen.

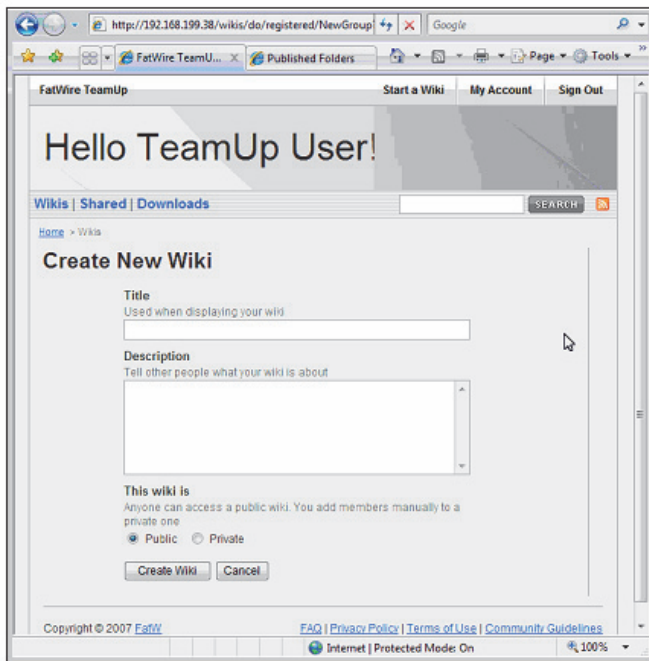


Figure 38. Oracle's TeamUp wiki module. Blogs and wikis are somewhat different from other micro-applications, since they require a WYSIWYG content contribution interface.

The downside to social applications within your CMS is that they may not represent best-of-breed functionality. For example, most blog tools that bundle with CMS packages do not have state-of-the-art, comments handling subsystems, complete with spam avoidance. That might not matter for intranet scenarios, but could present a rude surprise on a public website.

Alternatively, some CMS packages will “partner” with third-party providers of these tools, potentially offering the licensee a more robust package, at the expense, usually, of integration ease and a common interface.

Buyers should also note that portal software packages usually include these modules, as well. In fact, portal products and their ecosystems of third-party suppliers can typically supply a greater variety of these micro-applications, and they are more likely to be integrated tightly with the portal's access, security, and search subsystems — which is particularly critical for intranet scenarios. So licensees with portal software already on-hand face a strategic choice here.

Some Web CMS vendors are venturing into labeling their products as Social Content Management Systems (SCMS), exploiting SCMS as marketing buzz generator to deliver simple product additions. Buyers should look beyond the social-only focus and concentrate on the capabilities of the core platform and technology. Buying a CMS only because it has a Social CMS bow tied to it wouldn't be very shrewd.

Additionally, while the idea of UGC and community tools may not be extremely novel, you shouldn't underestimate the effort that goes into either building or supporting social micro apps. The very first question you should ask yourself when it comes to UGC and community tools is “Why do we need this?” For example, as noted earlier you need to be cautious about relying on your CMS vendor to deal with questions of spam in public environments. Media sharing is technically complicated as well, since it involves things like video transcoding to Flash, metadata management, addition of filters, etc.

Additionally, note that these are fast-moving technology segments, so you want to choose your suppliers carefully for adaptability. For example, in terms of ratings functionality, there is a trend of shifting from the number of stars as main indicators of good vs. bad to Facebook “likes” or an accumulation of Google+ +1s. All well and good, except that SharePoint — whose ratings are based almost indelibly on “stars” — cannot easily adapt to this transition, given its three-year-long development cycle. Can your CMS vendor?

Finally, you should recognize that these sorts of interactive applications bundled in your CMS *typically imply a reasonably coupled Production and Delivery architecture*. That is inherently neither a good or bad thing, but important to know beforehand. More specifically, you need to understand whether the delivery instance of your CMS can provide the sorts of “management” features you want on user-generated content, e.g. workflow, versioning, and audit trails.

Community and UGC Tools Requirements Builder

What sorts of applications do you need?

Generally, a poll or survey is simpler than a blog or wiki, which contain a richer store of information and require a real editorial interface. Data-driven applications like staff directories can be easily embedded within a CMS framework, but you will have to decide whether you want content managers actually updating source data through that interface, as well. Collaboration facilities like forums and document folders may require greater administrative oversight. In short, not all micro-applications are the same; determine your needs and evaluate software accordingly.

Do you need to put social media and social networking content under management?

Social media and networking content is very much in vogue right now, although most enterprises have not figured out exactly what sort of management they want to apply to this content — particularly if it is coming from (or going to) external sources. A handful of CMS packages allow you to put social and user-generated content into a formal workflow for internal vetting, but actual industry experience here remains quite thin.

Do you already have portal software?

If you already have a portal package in production, chances are good that you have access to most or all of these micro-applications as discrete portlets. For intranets in particular, it will likely make sense to use the portal's services rather than build them into the CMS.

Workplace Services: Supporting the Employee Digital Experience

While many WCM vendors target primarily externally facing sites, there remains a need for managing internal digital workplaces (formerly known as “intranets”). Clearly, content is important to the employee digital experience, so WCM technology remains germane. Likewise, some personalization features can work well both within the firewall and outside.

However, employees are looking for more than just content, and some WCM vendors have responded with additional services, like integrated dashboards and collaboration services. To be sure, most of these services traditionally lie more closely within the purview of enterprise portal and social/collaboration software. Because some enterprises elect to continue to build

their internal properties off the base of a WCM platform, some vendors in this segment offer services to support them here.

Collaboration and Social Networking Services

A WCM tool may provide a bevy of services to support internal collaboration and networking, albeit typically in a very lightweight way. On the plus side, Web CMS tools can usually bring some useful functionality to the social party, potentially including lifecycle, workflow, search, and library services.

Potential services here include:

- **Social Content Types** - Such as wikis, blogs, and micro-blogging (think Twitter within the enterprise). These services typically pale in comparison to what you could deploy from a best-of-breed vendor.
- **File-sharing and Project Services** - Sharing files is the mother of all contemporary employee collaboration, but most WCXM vendors provide only very crude folder structures for this. Likewise, you will likely look in vain for project management utilities.
- **Profile and Colleague Services** - These include a personal profile, perhaps a “wall” to aggregate activity streams, and the ability to “friend” or “follow” certain colleagues, and earn reputation badges. Don’t expect these services to be very sophisticated.
- **Discussion** - Many WCXM tools offer their own forums or can integrate decently with a third-party tool. What you’ll likely find missing is specific applications (e.g., communities of practice).

Dashboards: Exposing Key Data and Services

Not all, but some WCXM tools can offer a dashboard-type user experience, with multiple rectangular blocks of information and services. In the portal world, these go by various names, such as portlets, iviews, and Web Parts. In a WCXM tool, these tend to be simpler “elements” found in the UI that are optimized for obtaining content from their own repositories (and the occasional other source), such as traffic and analytics, task lists, and possibly a feed of collaborative comments.

In short, most WCXM tools can provide (or be customized to provide) a kind of activity dashboard that emphasizes systems and information the CMS controls. However, you may need to rely on custom extensions or other gadgets/widgets to expose data and services more broadly.

Ancillary Services: Beyond Content Management and Social Apps

In many scenarios, content managers need to do more than manage content. They need to manage the website as a whole, including a variety of corollary applications — some content-driven and some data-driven, but all part of the overall visitor experience centered on an interactive interchange of information. We addressed some of these applications in the Digital Marketing segment above, but there is potentially much, much more. Not the least of which is site search.

Some CMS vendors consider this sort of extended functionality as “out of scope,” while others provide various interactive tools natively as part of a full-blown site management package. As such, the enterprise-tier CMS packages will typically assume that you have licensed other applications already and want to take a best-of-breed approach. Conversely, on the lower and middle ends of the market, vendors will frequently assume that you want micro-applications included at least as options with your CMS. Increasingly, those options come in the form of supplementary modules of varying provenance supplied by the vendor’s “ecosystem.”

Site Search: How Users Really Find Content

On content-heavy sites, most visitors will turn immediately to a site search engine if they can’t find what they seek after one or two clicks. Indeed, a search engine often best meets the primary need of site visitors: rapid access to just the information they want.

However, a lot of hidden complexity can lie behind search engine implementations. Don’t make promises to users about relevancy rankings or the breadth of your indexes if you can’t keep them. If your search implementation cannot simultaneously query your catalog data and your unstructured documents, then you should provide users a choice of repositories via a pull-down menu to set their expectations.

Search engine results will quickly expose the quality of some of your metadata, including page titles and description tags. Great content with bad or missing titles is unlikely to be found by end users. The good news here is that your CMS should provide a way for you to apply the all-important title tag, description, keyword, and other custom metadata that will improve the quality of your users’ searches — as well as the presentation of the results. The CMS will also capture other implicit metadata, such as a document’s age, which can be very helpful for search results.

A good CMS can help your search engine in other ways. Ideally, the CMS package can be used to store and manage your taxonomy, helping you keep it fresh and accurate. A CMS can print a special, hidden sitemap of links just for your search engine crawler — kind of a private Google Sitemap.

At the same time, the imposition of a CMS may make your search effort more complicated and you will certainly want to retune any search engine after launching a CMS. It may also make your search efforts more costly, particularly when CMS user requirements — which can be fairly demanding — are taken into account. For example, newly empowered content managers working in a CMS will start modifying content structures in ways they couldn’t before. That’s generally a good thing, and in fact represents an important reason to implement a CMS. The good news is that you should be able to use that same search engine to find all these particularities, but you’ll want to check often.

In any case, you will want to customize and tune your search engine against your content repository. While you’re at it, you’ll have the opportunity to leverage other databased information (such as metadata), to make the index richer.

Some CMS vendors bundle site search services, usually Apache Lucene, but sometimes systems they developed themselves. For example, Zope comes with its own search engine (“ZCatalog”) to index and search its proprietary repository. However, it is sufficiently modular

that you can put other search engines on top of ZCatalog indexes, or alternatively, use ZCatalog to search other indexes.

You need to be careful here and test coupled search engines carefully against best-of-breed alternatives. As mentioned above, when your search engine is on intimate terms with your metadata constructs, you can generate dramatically better results. However, when a vendor has just applied its repository search service to your public site, the results are likely to be disappointing. Repository search often involves “advanced” fielded (or “parametric”) search, whereas site search usually entails simple text queries. Two different scenarios often call for two different types of search technology.

Online Forms: Many Uses for Experience Management

Online forms represent an important element in experience management strategy. First and foremost, web forms can be an effective way to enable your customers to communicate with you, while giving you the opportunity to gather precious data about them as potential leads, thus maximizing conversion rates. (Lead generation is one of the most common use cases for forms management in the context of web experience management.) Depending on how you segment your audience, forms can be personalized based on customer segments or scenarios.

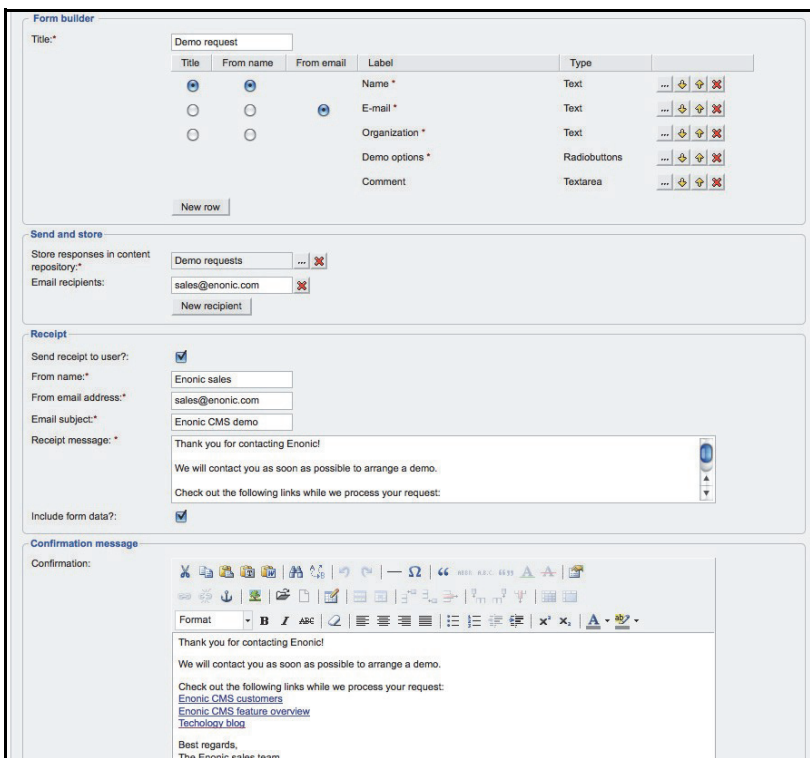


Figure 39. Form Builder in Enonic.

Challenges with online forms are plentiful. To begin with, not every CMS tool makes form creation user friendly. In many cases, you will still need to rely on a developer to create a form for you.

Forms on their own don’t provide much value if there’s no business logic behind them, or if they’re not integrated into a proper back-end or ancillary system (like your CRM or ERP package) to allow you to reap the most benefit in your experience management activities. An

integration that provides a two-way communication between a CMS and a CRM system for example, can prove to be much more meaningful, albeit not always easy to put in place.

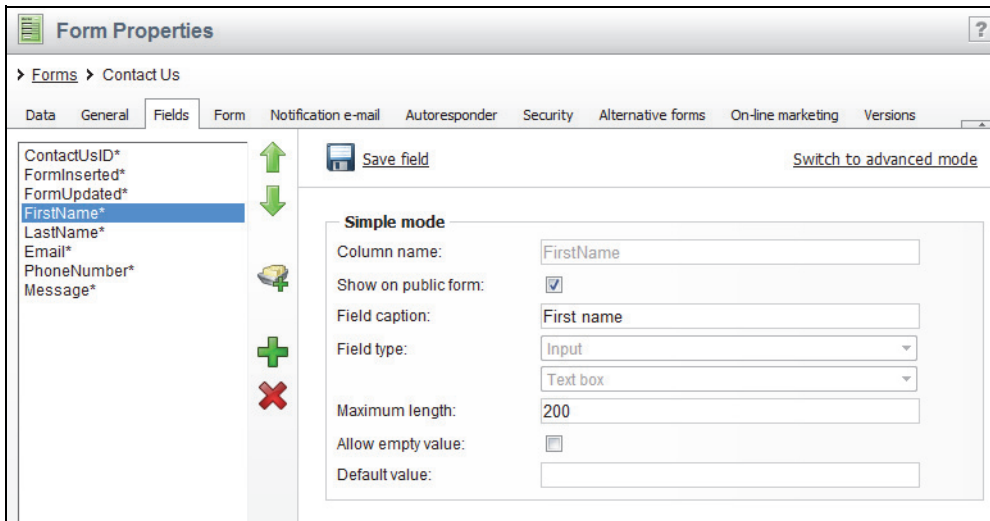


Figure 40. Online form field management in Kentico CMS.

You should examine licensing behind online forms modules carefully, since several CMS vendors still do not offer them as part of the core CMS but sell them as separate modules (for an additional price).

Module Ecosystem: Leveraging Additional Functionality

Web content and experience management (WCXM) platforms have increasingly evolved into frameworks for pluggable services and applications, and more product-oriented CMS tools have sought to offer “website-in-a-box” functionality through add-on services that we’ll call “modules.” This trend has pervaded both the commercial and open source communities. By extending CMS tools, modules can reduce development time, as well as align you with others trying to solve similar problems.

Modules can go by many names, including “widgets,” “plugins,” “extensions,” and “products,” but they do have a common purpose: more rapid application deployment. An add-on module can provide something as simple as re-rendering a default layout, but they more typically entail more complex functionality, involving reading and writing to a database or repository, with varying forms of interactivity. Some modules may comprise an entire service with multiple steps, e.g., a workflow subsystem or e-commerce environment, while others may encapsulate an individual task.

Modules can come from multiple sources, including the original vendor, consulting partners, other software companies, an individual developer or open source community, or even your own development team. That’s why we refer to a vendor’s module “ecosystem.” Many are free of charge; others cost money to license or to obtain support.

Modules bring obvious benefits, yet they often have hidden shortcomings. Here's a collection of potential problems you may encounter. If you have struggled with third-party portlets over the past decade, you may recognize most of these:

- Lack of standards, oversight, and certification
- Opaque provenance
- Poor quality
- Limited extensibility
- Interoperability problems and conflicts
- Lack of documentation and support
- Difficult upgrades

There is a common theme here: When you implement modules, you are suddenly working with multiple codebases — and potentially, multiple suppliers. This may constitute an improvement over developing substantial one-off code yourself, but to prepare to mitigate risks, you need to:

- Perform careful diligence
- Build time and resources into your project to research, review, test, and modify modules
- Plan for unit and integration testing
- Test to ensure that any module truly encapsulates your business logic (or can be extended to do so), and that it does so in a safe, effective, interoperable, and upgradable way

In most cases, you can turn to a platform “insider” (an individual, consultancy, or the platform vendor itself) that knows the module landscape and can steer you in the right direction, as well as provide the inside story of a module’s provenance and anticipated future. In short, be prepared to spend additional time and money.

Vendor Intangibles Dissected

Thus far, we have been discussing exclusively products and technology. However, the “fit” of a particular vendor to your enterprise needs, culture, and orientation is usually more critical to your overall success than the suitability of its CMS product.

In this section we introduce “vendor intangibles” — those other factors to consider when looking at a solution. In the individual product sections, we evaluate each offering according to four critical “vendor intangibles.” For specific advice on pricing and vendor selection, consult the final chapter in this report, “Part 5 – Advice, Pitfalls, and Best Practices” on page 836.”

Vendor Services

In the typical CMS project, customers spend more on services than software, sometimes many times more. To meet customers' consulting needs, most vendors maintain their own professional services organization (PSO). Most of the vendors in this report have also grown, to varying degrees, a "partner channel" of integrators, consultancies, and resellers who can help customers implement the package. We evaluate these two types of consulting services differently, because an offering is typically strong in only one type.

Vendor Professional Services

Like support, many factors can influence the quality of the implementation and other consulting services you receive from a vendor's PSO. Vendors often have "A" teams of their most top-notch implementation consultants — and then there's everyone else. (See advice in final chapter of this report about meeting the team that would work with you before you sign a contract.) "A" teams tend to work only on the biggest or most prominent client projects. "Everybody else" within a vendor PSO can include junior consultants, new inexperienced hires, and contractors that the vendor may or may not have quickly trained up on their tool. The vendor's own PSO will know the product best, but may be stuck in older ways of doing things and may not be up to speed — or even favor — the product's latest capabilities and modules. Of course, the vendor's PSO is highly unlikely to bring ancillary skills in areas such as information architecture or user experience.

Channel Partner Services

All vendors give at least lip service to "the channel," but in reality, some look to capture as much consulting work themselves as they can, and the depth and breadth of channel partners will vary dramatically from one vendor to another. It can also change over time.

The key for you, of course, is the *availability* of experienced services, ideally in some close proximity to your enterprise. Another factor in any services engagement is the consultants' experience in your industry, although this can be overrated (especially if you don't have a great need to integrate your CMS with other internal business systems). You may also want to know whether the integrator or reseller can bring suitable "soft" skills to bear — such as information architecture and graphic design expertise — in addition to just product knowledge.

We give better ratings to vendors that have larger, thriving reseller or integrator channels. Just remember that many partnerships are in name only. The depth of expertise in any given tool will vary wildly from firm to firm. An integrator may be very skilled in one of the vendor's products, but often not another. Perform as careful diligence of all consultancies as you would of the product itself.

Don't overlook the possibility of bringing in an independent developer or consultant that can also provide a useful source of knowledge and assistance. Products that have been available for sale longer can typically boast more experienced consultants out in the marketplace. Many of these consultants are alumni of vendor PSOs, which tend to have high staff turnover. However, the longer a tool has been out there, the higher the demand for these independent consultants, and so they cost more.

Remember that availability and quality of services matters just as much, if not more, for open source platforms, particularly since many of them tend to be complex to install and configure. As in the commercial world, there are sometimes tensions between the project “founders,” and other firms trying to make money providing services and support. You may end up having to work with a mixed team — core project founders plus local developers — much as you might with a commercial vendor. Commercial or open source, this places a premium on your ability to manage multi-supplier projects.

In almost all cases, there is a downside to selecting a “hot” vendor or product. Consultants who know the tool will be in high demand (and therefore scarce, expensive, and itinerant), as well. Without adequate expertise, you may lose the added advantage that the innovative technology was supposed to bring. So before you select a tool, make sure you know where, when, and how you are going to line up the necessary expertise to implement it properly.

Support & Maintenance

All technology buyers want to know “how well is this product supported?” All vendors rave about their product support. Nearly all customers complain about vendor support. What’s going on here?

Many factors can determine whether a vendor’s support will meet your needs. One key consideration is that vendors often offer different support mechanisms in different countries or locales, including different hours of support or varying expectations about how much first- and second-line support will be handled by a local partner. In general, the larger the vendor, the more variable you’ll find first-line support; many large vendors in this report have outsourced first-line support. That’s not necessarily a bad thing, but the person on the other line may never have actually performed an implementation of the product. Many small companies in this report will enable customers to reach “real” engineers in times of acute trouble. However, smaller vendors often have difficulty providing the 24/7 support that enterprise customers have come to expect. In almost all cases, you can spend more to get higher levels of support.

It is a fact of the software business that a vendor’s larger and/or more prominent customers will receive better support. Agreeing to take reference calls and speaking at user conferences also doesn’t hurt. Of course, you don’t necessarily want to be a vendor’s biggest customer, such that they don’t have the capacity to support you adequately. However, there will come a day when you will need to escalate a problem up through or beyond your vendor’s support department, and if you are a small fish in their pond, your likelihood of getting what you want accomplished drops substantially compared to your larger brethren. Customers who have contracted with their vendor’s professional services organization can sometimes later turn informally to the specialists who worked with them when a problem crops up (though you shouldn’t count on it).

In general, hosted “software-as-a-service” vendors tend to provide better support. One reason is that they’re on the hook for monthly fees from you, and are more likely to support your authors as well as your developers. (Traditional vendors typically ask you to support your contributors, while they support the software, itself.) In addition, they are providing a service, so the line between professional consulting and support becomes blurred. To that end, be sure that your support contract with a hosted provider spells out clearly what is tech support and what constitutes site enhancements.

Typically, a support contract with your vendor also includes maintenance. Examine the language here very carefully to be sure what it covers. Typically, it covers free software upgrades and patches, but the devil is in the details. Sometimes vendors will declare that a new major version of a product — typically after some major platform overhaul or change — actually represents a wholly new product, and that you therefore have to pay some fee for the pleasure of upgrading (or really replacement, in this case). Vendors will frequently release add-on modules at additional cost; sometimes these are snippets of code or subsystems that used to be part of the main product, which are now developed and marketed separately. Traditional vendor business practice holds that the best way to grow is to obtain new revenues from existing customers. Closely examine your maintenance contract accordingly.

Also remember that while maintenance covers the actual software patches and releases, it rarely covers the labor involved in migrating to the newer version. Be sure to budget for services here; depending on the scale and complexity of your implementation, this can take days or even weeks (sometimes months) to accomplish. This is another area where hosted vendors tend to have an advantage. Since they tightly control the environment, it is easier for them to upgrade customers — often without the customers even knowing it.

The Community

An increasingly important factor is the availability of authentic community support — including vendor developers, but also integrators and customers. Major open source projects really excel here, and only a few commercial vendors (notably Microsoft, but also HP and a handful of other suppliers) have seriously attempted to match the level of support ecosystem that you will find, say, in a Drupal or Plone implementation.

Don't underestimate the value of this kind of informal support. Vendor user-group meetings are also extremely helpful in this regard, giving larger or more geographically focused vendors additional leverage to compensate, perhaps, for their own, somewhat impersonal support channels. Developer extranets are also invaluable; the extent of these plays a big factor in our vendor assessments in this category.

Documentation

It's possible to generalize about the state of documentation of most CMS products in this report: It's bad. Among most open source projects and some commercial vendors, documentation is very, very bad. In general, you should not rely on the vendor's documentation to teach, or even explain, or even adequately document how a system works. Instead, invest in formal training (for both developers and contributors), and cultivating relationships with developers at the vendor, as well as other licensees of the product in your own industry.

Some vendors have acknowledged the difficulty in keeping their support documents up to date, and have developed various types of knowledgebases to capture and update more information in a useful, searchable way. Not all knowledgebases are very knowledgeable, but among some vendors, they are good enough that customers have learned to check there first. Request guest access and check it out before you sign any contract.

Strategy & Roadmap

“Strategy” is of course an abstract term and probably is overrated. Nevertheless, some vendors just seem to have a clearer and more passionate sense for what the market wants, and orient

their product development around specific objectives. Others develop software in more of a vacuum, and then see what sticks. Some vendors in this report just don't seem to understand the web. Others developed fine products five years ago, and milked them successfully, but don't seem to know where to go next.

Clarity of vision is more important to us than expansiveness. A vendor obsessed with becoming a winner across multiple scenarios is unlikely to satisfy any single customer very well. We therefore look for focus: Does the vendor have a clear idea of what use cases its product fits best? Has it structured its development accordingly? You should look for this too, but remember that vendor marketing messages (and open source project promotional efforts) tend to blur distinctions and create a false sense of omnipotence among vendor sales people.

Viability & Stability

If vision is one side of the vendor's organizational coin, then execution is the other. Buyers tell us they want stability and predictability from their software suppliers. In a fast-moving marketplace, that can be hard to find. We look at stability in very broad terms, across three general categories.

First is the general stability of a product or service in production. Does it hiccup regularly? Where? Why? All software is buggy, and some vendors have a habit of regularly releasing under-tested code into production. We try to isolate those tendencies in the individual product chapters.

Another issue is whether changes are underway or planned. The web is dynamic, so content management systems and technologies need to evolve, but migrations frequently become painful for customers. We tend to downgrade products here that are about to receive a major technology refresh. Vendors may find this unfair, but the last thing you want to do after spending half a year on a CMS implementation is go through a major upgrade (or wholesale replacement).

Of course, recently upgraded products are not always ideal candidates either. Some vendors will upgrade their tools to Java or .NET, thinking they understand those platforms well enough, when in fact, they often do not, particularly when it comes to performance. This doesn't mean you should avoid products that have been newly re-architected, but you should go in with your eyes open.

Finally, we look at vendor's financial viability and potential mergers and acquisitions ("M&A") disruptions and opportunities. This examination is necessarily subjective, but there are some clear markers, including cash position, sales trends, and whether any acquisition constitutes a good fit. Of course, you should not use this report as a substitute for your own diligence. The final chapter of this report ("Part 5 – Advice, Pitfalls, and Best Practices" on page 836) offers some guidance about how to conduct that diligence and select the right CMS package for your organization.

Universal Scenarios: The Key to Comparing Technologies

Let's look a bit beyond product functionality and vendor predilections. The right solution is the one that works for your needs. In our experience, the best way to understand your needs is through scenario analysis, which also provides the most efficient shortcut.

Explicitly or not, different CMS products target different use cases. Understanding the business scenarios that fit better or worse for the different packages enables you to see deeper into their relative strengths and weaknesses *for your particular circumstances*. Therefore, we have identified 11 common scenarios against which vendors can be judged. Each scenario emphasizes to varying degrees the features we elaborated in the previous section.

To be sure, these scenarios are abstractions. *In practice, your own web publishing effort is likely to represent variants or some hybrid combination of scenarios*. Additionally, the cases overlap somewhat. Nonetheless, they are useful for understanding what types of products tend to work better for each type of project.

Use this section — and the following chart — as a guide to identifying what vendors might fit your needs best. As you'll notice, some vendors fit well into multiple scenarios. Do not discount a vendor because we didn't list them in a particular scenario. Similarly, don't use this section to identify your short list of vendors for soliciting proposals. Rather, scan these charts to help identify your long list of suppliers to investigate deeper via the individual vendor chapters.

Overall, our 11 common scenarios for implementing a CMS revolve around 11 different types of web properties. We break them down into four broad categories. In trying to place your own needs, focus less on the category and more on the individual scenarios.

Simpler Site

- Informational
- Microsites & Landing Pages

Mid-Range

- Basic Digital Marketing
- Mobile Site
- Community Oriented

Complex Site

- Advanced Marketing Portal
- Global Enterprise
- Multichannel Publishing
- Ultra-Large Single

Digital Workplace

- SMB/Departmental
- Global/Enterprise

Simpler Site

These are small to mid-sized sites with reasonably standard features. This does not mean they are always simple to manage, per se, just that standard functionality required is relatively well known and widely available.

Informational Site

Though sometimes denigrated by interactivity snobs, basic corporate “brochure” sites play an essential role in conveying a brand and professionalism. Although perhaps only 10–100 pages in size, the consistency of message and overall look and feel is paramount. The informational site may be updated by only one or two marketing staff people. They do not need workflow, but typically require simple editorial interfaces with an emphasis on Word conversion, although they can learn the quirks of most systems. These editors tend to make changes to existing pages, with infrequent additions of new pages. The ability to roll back previous versions is very important, as is the ability to put changes into production on a moment’s notice, without IT intercession. Templates are typically developed by an outside agency and imported into the system by a part-time staff developer or external contractor. System architecture tends to emphasize static publishing for reliability, usability, and search engine optimization.

Informational Site	
Editorial Model	Tends to be highly centralized within marketing function, making infrequent updates
Content Model	Simple hierarchy: Page orientation with single body chunk
Functional Emphasis on	<ul style="list-style-type: none"> • Authoring & Transformation • Repository Services
Typical Adopters	<ul style="list-style-type: none"> • Small to mid-sized businesses (SMBs) and non-profits • Parent holding companies
System Modifications	Infrequent, with configurations set by novice IT staffer or outside consultant

Microsites & Landing Pages

This scenario envisions an enterprise that must regularly create small sites in connection with a specific issue, event, or partnership. For marketing campaigns, these may simply be stripped-down landing pages, typically with a form or some other call to action.

Microsites and landing pages may vary greatly in content and profile of contributors, but the enterprise typically wants to maintain some modicum of central control over templating and structure, perhaps with the ability to cascade changes from a central instance to multiple microsites. Role Management, and the ability to delegate authorizations, becomes important, as does Templating (for nested control of layout) and Promotion (for pushing content to

remote web servers when necessary). The ability to stamp out or “clone” sites and settings from a central model makes creating microsites much simpler.

Microsites & Landing Pages	
Editorial Model	Tends to be highly decentralized, including contributors from beyond the enterprise
Content Model	Semi-structured: Limited set of content types
Functional Emphasis on	<ul style="list-style-type: none"> • Authoring & Transformation • User Interfaces • Tagging & Taxonomy • Repository Services • Templating • Content Reuse • Site Search • Site Analytics • Forms Management
Typical Adopters	Mid-sized to large enterprises and non-profit organizations
System Modifications	Infrequent modifications revolve mostly around tweaks to content types and templates; ideally possible through a browser interface

Mid-Range Site

Most organizations eventually move “beyond the basics” with requirements that drive a more engaging, interactive, and overall richer experience. At this stage, organizations are looking to tap into digital marketing initiatives that go beyond microsites and landing pages, and into more interactive experiences around communities, as well as initial steps in testing & optimization, and analytics. Basic mobile content management and delivery are also often explored here.

Basic Digital Marketing

This describes an interactive, data-driven site that employs standard applications but does not entail unusual functionality that would require advanced development. Some of these applications might include those found in the “Community” scenario, but they are publisher-driven rather than visitor-driven, and tend to focus on persuading the visitor to engage. Also, site owners may want to enable multiple ways to navigate to the same content, typically through basic metadata and classification. This scenario may entail some basic database query interfaces, exposed through the CMS template.

Basic Digital Marketing	
Editorial Model	Tends to be fairly centralized
Content Model	Semi-structured: Limited set of content types

Basic Digital Marketing	
Functional Emphasis on	<ul style="list-style-type: none"> • Authoring & Transformation • Tagging & Taxonomy • System Reporting • Templating • Limited Digital Marketing Services • Forms Management • Module Ecosystem • Analytics • Testing & Optimization
Typical Adopters	<ul style="list-style-type: none"> • Small- to mid-sized businesses and non-profits • Small media firms • Local governments
System Modifications	Semi-frequent tweaks and a trained developer is required to customize and enhance system

Mobile Site

In some cases, you will want to eschew showing your “regular” website on tablets or smartphones and instead create dedicated mobile websites or even content-driven, device-specific mobile apps (e.g., for Android or iPhone). As discussed earlier, this can be most simply addressed via mobile templates and specialized CSS, but many organizations want to take it to the next level with a dedicated mobile experience that’s quite distinct from their default web experience. This often requires both specialized content models and mobile-specific coding and media production.

Mobile Site	
Editorial Model	Tends to be fairly centralized
Content Model	Semi-structured: Limited set of content types Highly structured: often fine-grained
Functional Emphasis on	<ul style="list-style-type: none"> • Authoring & Transformation • Content Modeling • Content Reuse • Templating • Mobile Services • Module Ecosystem • Analytics

Typical Adopters	<ul style="list-style-type: none"> • Small- to mid-sized businesses and non-profits • Small media firms • Local governments
System Modifications	Semi-frequent tweaks and a trained developer is required to customize and enhance system

Community Oriented

Clubs, associations, civic groups, professional networks, and other membership organizations frequently need an online destination to communicate and share opinions and information. More than just a simple forum, the site is typically owned and managed by the community leader, who may publish updates and other unstructured content. However, the core of the system is specialized content that is usually behind a pay wall or registration/login and may include wikis, threaded discussion, blogs, photo sharing, chat, polls, surveys, content rating, and other interactive services. At some level, these sites are really small portals, but most portal software — especially commercial portal packages — are too complex for this purpose, while some CMS tools target this scenario explicitly. In addition to providing a variety of authoring services, this scenario emphasizes Role and Group Management, to be able to distinguish carefully among different classes of visitors and contributors.

Community Oriented	
Editorial Model	Tends to be highly decentralized and in fact, may rely substantially on community (a.k.a., visitor) input
Content Model	Highly unstructured: Limited set of content types
Functional Emphasis on	<ul style="list-style-type: none"> • Authorization and Authentication • Authoring & Transformation • Repository Services • Site Search • Mobile • Digital Marketing Services • Social Apps • Module Ecosystem • Community & UCG
Typical Adopters	<ul style="list-style-type: none"> • Clubs • Associations, professional networks, and other membership groups • Major media firms
System Modifications	Semi-frequent tweaks and configurations are set by novice IT staffers or technical webmasters

Complex Site

In the realm of complex scenarios, the sites are just as much applications (or sets of interconnected applications) as they are collections of content. Page counts frequently rise into the tens to hundreds of thousands, and in the case of very large product-based companies or multinational enterprises, perhaps even number more a million. Oftentimes, companies that start with the more mid-range scenarios evolve into needing a suite of more complex ones (such as a global digital workplace) that must be tied together. This creates a need to accommodate more complex workflows, governance structures, and content models.

Advanced Marketing Portal

This scenario envisions an enterprise (mid-sized and up), focused on the web as a marketing channel, typically in conjunction with specific e-commerce, sales, advocacy, or other persuasive efforts. Marketing staff are dedicated to electronic dialogue with prospects, members, customers, or other constituencies via diverse electronic channels, including email, syndication, and mobile. Although role management remains trivial and workflows themselves tend to be simple, marketing managers must be able to vet and assemble compound “packages” of information that include multiple types of content items (images, text, and rules) to be published at a later date. Contextual delivery of web content with attendant needs to test and measure becomes paramount. In particular, marketers need to establish discrete promotions and set diverse personalization rules, availing themselves of profile-, behavioral-, and preferences-based alternatives. This makes these sorts of properties more “portal-like,” and in some cases customers may want to review enterprise portal software itself as a plausible base for their solution. In any case, managers must be able to modify important configurations themselves (e.g., the order of elements in a template or on a page — if not the template, itself).

In some cases, the system must make provision for visitor-generated content. Polls, surveys, commenting, feedback, and other portal-like services may be required. Complex content models and diverse permutations put a premium on fully virtualized preview. The product’s metadata model must support multifaceted taxonomies and allow for simple updates to corporate vocabularies. The system may integrate with an enterprise marketing automation, CRM, or e-commerce system, and almost certainly with an email marketing service. The architecture supports a dynamic delivery model, with an emphasis on component caching, where possible.

Advanced Marketing Portal

Editorial Model	Tends to be highly centralized within marketing function
Content Model	<ul style="list-style-type: none"> • Fine-grained content chunks, often object oriented • Contextual content assembly • Multifaceted taxonomies

Advanced Marketing Portal

- | | |
|------------------------|--|
| Functional Emphasis on | <ul style="list-style-type: none"> • Authoring and Transformation • Tagging & Taxonomy • System Reporting • Templating • Personalization & Segmentation • Multichannel & Mobile • Media Management • Dashboards • Testing & Optimization • Site Analytics • Community & UGC • Social Media Integration |
|------------------------|--|

-
- | | |
|------------------|---|
| Typical Adopters | <ul style="list-style-type: none"> • Consumer goods / services companies • Large advocacy organizations |
|------------------|---|
-

System Modifications	Constant, and major modifications are undertaken by dedicated, trained developer staff; marketing managers tune settings daily via browser control panel
----------------------	--

Global Enterprise

This scenario focuses on large organizations that may have multiple subsidiaries, divisions, chapters, or brands under one umbrella. Some content may be managed at the enterprise level and then shared among subsidiary sites, so content and templates reuse is paramount. Even more challenging is the occasional need to publish slight variations of content and layouts based on the brand or sub-site. In many global enterprises, multiple WCM systems are in place as a result of disparate IT departments and budgets, and only recently has there been an effort to consolidate and centralize both the technology and the process of managing content.

As in other complex scenarios, ideally a central governance council at the corporate parent company sets rules, but more often than not, subsidiaries or departmental sites are given a good deal of leeway to operate independently. Localized user interfaces for content managers, worldwide training, and system replication are core to making content management feasible on a global scale. Globalization support and parallel workflow capabilities are necessary in this scenario, as content often needs to be localized as part of a complex, multinational workflow. Multiple instances on numerous servers are usually required to handle load and traffic.

Global Enterprise

Editorial Model	Tends to be somewhat decentralized across brands and subsidiaries, sometimes with a central governance council, but few centralized workflows
-----------------	---

Global Enterprise	
Content Model	<ul style="list-style-type: none"> • Varies greatly based on the nature of the company • If product based, deep hierarch — often separate ones for product data and content • Structured product data: Semi-structured content
Functional Emphasis on	<ul style="list-style-type: none"> • Role Management • Authoring & Transformation • User Interfaces • Repository Services • Tagging & Taxonomy • Globalization • Reporting • Scaling & Performance • Personalization & Segmentation • Site Search
Typical Adopters	<ul style="list-style-type: none"> • G2000 and other large, multinational enterprises
System Modifications	Infrequent at enterprise level (configurations determined by governing body and implemented by dedicated IT team), but frequent at departmental level, with diverse internal and external resources

Multichannel Publishing

In Multichannel Publishing, the emphasis is on repurposing content — that is, taking the same piece of unchanged content and publishing it to multiple channels and formats. An example might be a stock quote published on a website, mobile application, a TV device on the back of an airplane seat, on a terminal in a call center at a brokerage house, and finally, in a PDF mutual fund report. Single-sourcing content is the ideal in this scenario, thus there’s heavy emphasis on not only the versatility and granularity of the content model, but also the format in which content is stored.

Deployment of content is tricky in this scenario, as any content update requires cache clearing and/or refreshing of the end publishing mediums, which could number in the dozens. Oftentimes, this is not handled by your CMS, but an integration with a portal, application server or other delivery mechanism. If content needs to be localized and then redeployed across channels, the process becomes even more complicated and system synchronization and replication are often necessary to keep the data flowing fluidly.

Multichannel Publishing	
Editorial Model	Highly centralized, with a central governance council and workflow
Content Model	<ul style="list-style-type: none"> • Deep hierarchy • Highly structured

Multichannel Publishing

- | | |
|------------------------|--|
| Functional Emphasis on | <ul style="list-style-type: none"> • Authoring & Transformation • Aggregation • Tagging & Taxonomy • Workflow • Repository Services • Templating • Personalization & Segmentation • Multichannel • Mobile • Social Media Integration • Community & UGC • Promotional Campaigns • Online Forms |
|------------------------|--|

-
- | | |
|------------------|--|
| Typical Adopters | <ul style="list-style-type: none"> • Publishing companies • Media outlets • Technical documentation teams |
|------------------|--|

System Modifications	Infrequent, and configuration changes and system modifications can carry significant downstream implications
----------------------	--

Ultra-Large Single Site

In this scenario, scalability — both in terms of performance and functionality — becomes paramount. With site pages numbering in the hundreds of thousands up to and over a million, the number of content contributors might be in the dozens to hundreds, while the number of end users may rise into the millions. Altogether, this creates a great deal of stress and load on the system. Emphasis is on the ability of dozens of content contributors to be logged on at once and not experience time delays with authoring and repository search. Perhaps the most common Ultra-Large Single Site scenario is the large media site or portal, where hundreds of articles on various topics are published daily, and linked together based on metadata.

Content is usually varied and highly relational, taxonomies numerous and multifaceted, and often the majority of the content is not placed by a content manager, rather it is tagged and then the system calls it up dynamically and in multiple locations on the site. Thus, features such as content preview become vital in order for content managers to understand the consequences of their tagging. System reporting, batch workflows, and syndication are all core pieces of functionality for the Ultra-Large Single Site to be successful.

Ultra-Large Single Site

Editorial Model	Tends to be highly decentralized, with specific content owners for site sections with a central governance council but few centralized workflows
-----------------	--

Ultra-Large Single Site	
Content Model	<ul style="list-style-type: none"> • Simple but very deep hierarchy • Page orientation, but multiple, task-oriented paths to the same content
Functional Emphasis on	<ul style="list-style-type: none"> • Role Management • Authoring & Transformation • User Interface • Repository Search • Tagging & Taxonomy • Workflow • Reporting • Scaling & Performance • Site Search • Multichannel • Mobile • Archiving & Compliance • Site & Campaign Analytics • Social Media Integration • Testing & Optimization
Typical Adopters	<ul style="list-style-type: none"> • Major consumer brands • Media portals • Large public agencies
System Modifications	Somewhat frequent, based on new content types or “special packages” of content, and configurations are determined by a governing body and implemented by a dedicated IT team

Digital Workplace

Digital workplaces for your fellow employees — a.k.a., intranets — are fundamentally different platforms than public websites. In addition to lighter security requirements, they tend to emphasize different services, such as document management and possibly social/collaboration services (unless those come from another system). Intranets can also require similar services that you see on a public site — such as workflow and personalization — albeit implemented in very different ways.

SMB/Departmental Workplace

The small, single-site intranet can become very complex, but most aren't. Document sharing and the ability to traverse directory indexes starts to become more important, although for more advanced document collaboration, a system like Microsoft SharePoint may be more apt. For richer workplace experiences beyond simple file sharing, however, a CMS can provide useful services and certainly simplify the process of contributing unstructured text. Editorial

usability begins to rise to the fore, as authorship is distributed beyond the communications department to individual webmasters or contributors.

SMB/Departmental Workplace	
Editorial Model	Tends to be somewhat decentralized
Content Model	Semi-structured: Limited set of content types
Functional Emphasis on	<ul style="list-style-type: none"> • Authoring & Transformation • User Interfaces • Tagging & Taxonomy • Repository Services • Document and Media Management • Collaboration & Networking • Site Search
Typical Adopters	<ul style="list-style-type: none"> • Mid-sized businesses and non-profits • Departments within larger enterprises • Call centers and others relying on internal knowledgebases
System Modifications	Semi-frequent tweaks and a trained developer is required to customize and enhance the system

Enterprise / Global Digital Workplace

An Enterprise Digital Workplace spans multiple departments, and in fact, may constitute multiple websites and a myriad of applications. In some cases, the WCXM tool must push content to or through a portal application, as well as support freestanding websites. With contributors potentially numbering in the hundreds, distributed Role and Group management becomes critical, as does the ability to create specialized interfaces for different classes, users, or separate tasks. Unlike smaller workspaces, the enterprise level requires much more sophisticated template management, with some degree of centralized design control with limited departmental flexibility. Also, at this level, page generation, and performance more generally, can become an issue. Finally, to the extent that original content is being created (and not just exposed from other repositories), the enterprise needs some sort of strategy for records management.

Global enterprises may come to see the value of a truly Global Digital Workplace — perhaps initially one that resides “above” separate regional sites — with the long-term goal of consolidating local content on a single platform. A central governance council sets rules and substantial “global” content originates from headquarters, but with the exception of the uppermost pages, content tends to be managed locally. Localization and parallel workflow capabilities become paramount here. Overall, content management frequently migrates to a set of information specialists in different countries who spend ample time in the system and have less of a need for novice interfaces.

The need for limited look-and-feel adjustments to local content favors templating schemes that are object oriented. The ability to define custom roles for different regions also becomes important. Simple profile- or preferences-based personalization is often required, although the

enterprise may move this function to a portal package. In any case, ease of integration with a current or prospective enterprise portal solution also becomes critical — unless the WCXM platform can provide sufficient portal services.

Enterprise / Global Digital Workplace	
Editorial Model	Tends to be somewhat decentralized
Content Model	Semi-structured: Limited set of content types
Functional Emphasis on	<ul style="list-style-type: none"> • Authorization & Authentication • Authoring & Transformation • User Interfaces • Tagging & Taxonomy • Repository Services • Mobile • Collaboration & Networking • Dashboards • Globalization • Reporting • Personalization & Segmentation • Site Search • Archiving & Compliance
Typical Adopters	<ul style="list-style-type: none"> • Mid-sized to large enterprises • Enterprises with a global footprint
System Modifications	Frequent modifications, with dedicated, trained developer(s) required to customize and enhance the system

Did this section help you? How could we make this section more useful? Write to us at feedback@realstorygroup.com.

Part 4 – CMS Product Comparisons

Overview of the Landscape: Growth and Some Specialization

In your search for a content management system, you will find an increasingly confusing array of technology players. There are thousands of systems in the marketplace that claim to offer WCM capabilities. Out of these, all but perhaps 100 are really the stepchildren of small, regional integrators and consultancies.

Growth

As web content expansion hit critical velocity in the late 1990s, large and small firms alike sensed the vacuum described in “Part 2 – Content, Experience, Management, and Systems” of this report and leapt into the fray. Some large firms, like Microsoft, Adobe, and IBM didn’t want to risk a hole in their product line, and acquired CMS technologies from other firms.

Other companies converted one-time applications that were developed in-house (or as a result of consulting work) into full-blown commercial products. This trend continues, as small integrators seek to bring their own offerings to market as semi-packaged software that has various levels of maturity from its “agencyware” roots. (We review this class of vendors in the section “Regional and Niche Players” on page 834.)

At the same time, an evolving WCM marketplace has seen the advent of SaaS vendors, which started appealing to customers without large IT departments. Open source CMSs have sprung as well, sometimes promising “free” software, but more generally promoting a well-suited development model for many IT departments. More than fifteen years since its inception, the field of WCM continues to grow.

Some Specialization

The nearly ubiquitous marketing jargon that obscures true product niches compounds the dilemma for would-be CMS buyers. This section of the report is designed to provide you with an accurate roadmap of the current CMS product landscape.

Real Story Group selected these products because they are indicative of the type of offerings available in their class. They were not chosen because we believe they are inherently stronger, weaker, or more popular than others are in their same group.

Assessing Different Delivery and Licensing Models

In the segments that follow, we’ll throw around terms like “SaaS,” “Open Source,” and “Dual License.” In this section, we’ll define those terms and investigate the pros and cons of those approaches.

The Hosted, “Software as a Service” Model

As centralized corporate IT efforts have waxed and waned, there is clearly a trend to push budgets and decision-making power down to the departmental level in big companies. This helps hosted CMS providers. If your company owns an enterprise license for one of the “Majors,” but does not intend to implement it in your department anytime soon, a hosted (once

known as “ASP” and now typically called “SaaS” — software as a service) solution might start looking favorable.

We have found that — with no need to worry about regression testing new features for various heterogeneous platforms — hosted CMS vendors tend to roll out new features to the entire installed base more frequently than their server-based competitors. While you may not need to worry about upgrades in this case, in almost all cases, upgrades happen automatically. Some customers do not appreciate the “all or nothing” approach to this type of new product version rollouts, because they prefer tighter control over introducing new functionality to their user base. This becomes a real issue when a customer has done several customizations to various CMS components, because this often exacerbates automatic upgrade problems.

Sometimes this differs little from enterprise-tier content management systems, where in some cases more than half of all customers have not upgraded to the latest major release, because previous customizations left them almost hopelessly forked off the main development path of the product, essentially making “upgrades” a complete do-over.

In any case, you will need to gauge your comfort level with outsourcing your web production and publishing platform. If your web efforts represent a true cost center and you are unlikely to need to integrate your content management system with other corporate systems anytime soon, hosted alternatives could provide an economical approach to enhancing your publishing capabilities without expanding staff and infrastructure.

The other reason to consider a hosted vendor is if you need to launch a CMS in a relatively short period. A hosted vendor can often get you up and running with a full-blown system (not just a prototype) in a matter of weeks or months (depending on the complexity of your requirements), because they don’t have to expend time and effort on installing and configuring servers in your particular (read: quirky) environment.

Note that hosted approaches are less likely to be well suited to intranet/digital workplace projects, since SaaS CMS providers may have limited capabilities to integrate (especially integrate real time) with your other internal applications, such as your ERP or HR systems. For straightforward web publishing, this is usually not a big issue. For complex page assembly from multiple repositories, you should test carefully.

Finally, as the hosted model gains traction, several traditional CMS vendors (like SDL, Oracle, Ektron, EPiServer, and others) have come out with hosted versions of their offerings, often in partnership with a consulting or hosting company, but sometimes in their own data centers. If you prefer one of these “traditional,” on-premise tools but want to buy it as a service rather than a product, their hosted alternative may well be worth investigating. We would just caution you on the following:

- Traditional products are not architected from the ground up to run as hosted solutions, and you may experience some glitches running them in a shared, multitenant environment; in a dedicated environment, you are just buying managed hosting with potentially some other management services on top
- You likely will be dealing with two companies (the software vendor and its hosting partner), rather than one

The main benefit of running an application in a “cloud” environment is scalability and elasticity. You may never reap this benefit with a traditional CMS that lacks proper architecture for cloud services.

Community Open Source

Open Source software always seems attractive on the surface, because of the perceived absence of cost. It is important, however, to distinguish among various licensing options that are now available in open source. Even if you do not need to pay anything for the license itself, more often than not, community support & maintenance, additional modules, or enterprise-ready functionality will be hidden behind some sort of a payment plan. With the increasing commercialization of open source (read on the next section on commercial open source), OSS vendors — and sometimes community members — will offer hosted services and professional services for an additional fee. While open source has never been free, lately you'll find myriad ways to spend money on related services and capabilities; therefore, don't be under the impression that selecting an open source CMS will mean an absence of monetary investment on your part.

Finally, remember that there's a difference between true open source and freeware. The latter encompasses the likes of freely distributed CGI scripts. By contrast, open source projects are true community efforts to create software that equals or exceeds commercial rivals. The Linux operating system and unparalleled Apache web server are two poster children for the open source movement.

“Open Source” describes a license that meets a set of criteria defined by the Open Source Initiative (OSI). The OSI certifies over 50 licenses as being open source, but some licenses are more restrictive than others are. It is important to understand the licensing terms of the open source software that you use because that will determine what you can legally do with it and your obligation to the community.

Evaluating open source software is easier than evaluating commercial software in some ways and harder in others. Community open source software selection is harder, because there is frequently no software sales organization to help you understand the application and its potential. However, open source software and the communities behind it are more transparent than commercial software. The information is out there and, if you know where to look, you can really understand the quality of the software, and the helpfulness and strength of the group that supports it.

What to Look for in an Open Source CMS

We urge you to look for open source projects with velocity and staying power. That way, if the original evangelists move on, a critical mass in the community will keep a project going. A lively and active worldwide open source project can expose you to some of the best and fastest technical support found anywhere. We encourage you to read the community mailing lists for the quality of the responses and who is posting them. Look for projects with many people posting helpful answers, not just a few. Also, read the mail lists to get an idea of what other companies are doing with the software. A community of organizations with similar scenarios will be more likely to extend the application in ways that are useful to you.

One reason why project velocity and staying power matter is that open source CMS packages actually require substantial integration. They tend to take a platform orientation. This is beneficial at many levels, because it enables you to take advantage of other open source tools (e.g., ad management or catalog tools in PHP or Python) as part of a broader solution. Additionally, since they typically adhere more closely to industry standards than commercial

alternatives, you shouldn't write off enterprise integration either, but similarly, don't look for prebuilt ERP integrations.

Since the major open source CMS packages grew out of application server environments, they tend to be stronger at delivery and weaker on production. Indeed, they often leave much to be desired with respect to usability and end-user features. This begs a deeper question: "Will these developer-driven solutions ever appeal to non-technical users who increasingly influence — if not dominate — the CMS buying process?" To be sure, some buyers will gravitate to open source packages, solely because of the licensing and support models. This is especially true in higher education. On the whole, though, the transition from developer platform to marketable software remains a huge challenge, and not surprisingly these packages tend to get adopted in enterprises when there are more technical users (like webmasters) than business users.

Consider an open source solution if you have highly custom needs, a strong IT staff, and the patience and motivation to develop usable interfaces with your own resources. The good news is that a typical open source package will give you the freedom to do just that.

Commercial Open Source

Some open source projects are owned and managed by commercial companies that provide professional support and drive the development of the platform. These companies occasionally distribute their software under both open source and commercial licenses to offer a more traditional software buying option for their customers. In other cases, the companies make money by serving as the default choice for support contracts (and integration, as well).

The level of community contribution in commercially driven, open source projects depends on the strategy of the company. In most cases (like DotNetNuke, eZ Publish, and Hippo), the core application is managed by the company and the community develops add-on modules.

In other cases (like Magnolia), only the commercial version is officially supported. Note that support fees can become quite steep, equaling (or even exceeding) the support and maintenance fee for a comparative commercial offering.

On the whole, though, most of the same criteria apply here as with "pure" open source solutions. You will want to look for a broad development community and a history of steady improvements and support.

Because of the dominance of a single vendor providing support, consulting, and controlling the code base, you should recognize that the communities around these tools tend to be less diverse and vibrant than open source projects managed by an independent foundation supported by a plethora of different firms.

Examining Basic Specifications

Some basic specs and characteristics about a vendor can and should shape your short list. In what areas of the world is the vendor active? What operating systems does the product support? What are its core strengths and weaknesses? What other potentially similar vendors should we consider?

The following chart therefore kicks off each vendor chapter.

Specsheet	Product Summary
Geography	In what regions of the world is the product most actively marketed?
What's New	What has changed with the offering since the last edition of this report?
Strengths	What is the package generally good at doing?
Weaknesses	What is the package generally weak in doing?
Potential Fit	Which scenarios are the best target for this package?
Unlikely Fit	Which scenarios generally don't thrive with this package?
Compare to	What are some analogous products, including those in other tiers?
Operating System	On what Operating Systems does the product run?
Repository	Which databases or other repositories will it employ?
Client	How do content contributors and administrators access the system?
App Platform	What development platform does the product use?
Licensing	How is the product priced?
Ownership	Is the company publicly or privately held?

How We Categorize Vendors

Each individual vendor has its own story, but we try to break the tools into categories to help you understand where a particular vendor — or product — is going.

Product Categories by Tier

For comparison purposes, our analysis splits the vendors into the following categories:

- Legacy Platforms
- Upper-range Platforms
- Mid-range Platforms
- Mid-range Products
- Simpler Products

Each category is described in greater detail just before the relevant product evaluations, but in the meantime, here is a short “cheat sheet” to help you narrow your choice by technology, geography, and other factors.

	Primary Geography	Primary Delivery Model	Application Environment	License Type	License Model
Legacy Platforms					
HP	Global	On premise	Hybrid: C++/Perl/JS/Java	Commercial	Based on seats, server modules
IBM	Global	On premise	Java	Commercial	Based roughly on CPUs
OpenText	Global	On premise	Java	Commercial	Based on CPUs or “installation”
Oracle	Global	On premise	Java	Commercial	Based on CPUs and modules
Upper-range Platforms					
Adobe	Global	On premise/Cloud	Java	Commercial	Based on servers and modules
CoreMedia	Europe, esp. Germany, some US, Asia	On premise	Java	Commercial	Based on CPUs and seats
Percussion	Global, esp. North America, UK	On premise	Java	Commercial	Based on servers
SDL	Global, esp. North America, Europe	On premise	Hybrid: COM, .NET, Java	Commercial	Based on servers and modules
Sitecore	Global, esp. North America, Nordics	On premise	.NET 3.5	Commercial	Based on servers, modules, users
Mid-range Platforms					
Atex Polopoly	Global	On premise	Java	Commercial	Complex, variable mix of factors
Drupal	Global	On premise/SaaS	PHP	Open Source	GPL
EPiServer	Nordics, North America, South Africa	On premise	.NET	Commercial	Based on sites
Hippo	Europe, esp. Benelux, some North America	On premise	Java	Dual	Open source; commercial per CPU
Microsoft	Global	On premise	.NET	Commercial	Based on seats (internal), CPUs (ext.)
Plone	Global	On premise	Python/Zope	Open Source	GPL

	Primary Geography	Primary Delivery Model	Application Environment	License Type	License Model
TYPO3	Global, esp. Northern Europe	On premise	PHP	Open Source	GPL
Mid-range Products					
CrownPeak	North America	SaaS	COM	Commercial	Based on site size, users
E-Spirit	Europe, some North America	On premise	Java	Commercial	Based on implementation, modules
Ektron	Global	On premise/ Hosted	.NET	Commercial	Based on URLs, contributors
Enonic	Nordics, esp. Norway	On premise	Java	Commercial	Based on CPUs
eZ Publish	Global, esp. Northern Europe	On premise	PHP	Dual	Open source; server licenses
Hannon Hill	North America	On premise	Java	Commercial	Based on servers or CPUs
Ingeniux	North America & Europe	On premise / SaaS	COM/.NET	Commercial	Based on servers
Upland Software	North America	SaaS	Java	Commercial	Based on users, websites, and traffic
Magnolia	Europe, some North America and Asia-Pac	On premise	Java	Dual	Open source; server licenses
CCI (Escenic)	Europe, esp. Nordics	On premise	Java	Commercial	Complex, variable mix of factors

	Primary Geography	Primary Delivery Model	Application Environment	License Type	License Model
Simpler Products					
Automatic WordPress	North America, Europe, Australia, and Japan	On premise	PHP	Open Source	GPL
DotNetNuke	Global, esp. Europe, North America	On premise	.NET 2.0	Dual	Open source; support per instance
Joomla!	Global	On premise	PHP	Open Source	GPL
Kentico	Europe, some North America	On premise	.NET	Commercial	Based on servers and users
MODX	Global, esp. North America	On premise	PHP	Open Source	GPL
OmniUpdate	North America	SaaS / Appliance	PHP	Commercial	Based on sites
OpenCms	Global, esp. Europe (Germany)	On premise	Java	Dual	Open source LGPL and commercial modules
Salesforce	Global	SaaS	Force.com	Commercial	Based on users & websites
Telerik	North America, Europe, Australia	On premise	.NET	Commercial	Based on domains
TERMINALFOUR	UK, Ireland, some North America	On premise	Java	Commercial	Based on content items

Matching Features to Requirements

To help you compare and evaluate products based on your requirements, we will use the following charts from the feature set discussion in the previous section. At left is an analysis of product features by lifecycle phase along with our list of “vendor intangibles,” with accompanying explanatory key (right).

Phase / Attribute	Rating	Key
Technology		○ Product does not provide this feature in any meaningful way.
Technical Administration & Security		◐ Product provides this feature, but is not as mature as its rivals
Threat Prevalence	◐	◐ Product provides this feature
Authentication & Authorization	◐	◑ Product excels at this feature, <i>relative to other products in the same category</i>
System Reporting	◐	● Product masters this feature, <i>relative to other products in the same category</i>
Multisite Management	◐	
Cloud Services	◐	
Development		
Configuration & Customization	◐	
Integration & Extension	◐	
Content Modeling	◐	
Templating	◐	
Performance		
Back-end Performance	◐	
Site Caching & Delivery	◐	
Content		
Contributor Experience		
Overall Usability	◐	
UI Accessibility	◐	
Contributing Content		
Authoring & Transformation	◐	
Tagging & Taxonomy	◐	
Content Reuse	◐	
Media & Document Management	◐	
Repository Services	◐	
Content Lifecycle		
Workflow	◐	
Globalization	◐	
Archiving & Compliance	◐	

Phase / Attribute	Rating
Experience	
Publishing	
Standards Adherence	
Multichannel	
Mobile	
Digital Marketing	
Site & Campaign Analytics	
Testing & Optimization	
Segmentation & Personalization	
Social Media Integration	
Promotional Campaigns	
Community & UCG	
Workplace	
Collaboration & Networking	
Dashboard	
Ancillary	
Site Search	
Online Forms	
Module Ecosystem	
Intangibles	
Digital Marketing	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

Putting Scenarios and Vendors Together

Having described 11 universal scenarios in the previous section, let's chart which vendors appear to be better or worse "fits" for each one. Note that, like all Real Story Group rankings, vendors will offer you a different story or suggest that our assessment of their suitability for different scenarios is too restrictive. You should remain skeptical. A vendor that tries to be good at many things is unlikely to excel at any. As it happens, nearly all vendors in this report have a particular sweet spot. The following chart shows where.

Putting Scenarios and Vendors Together	Simpler Site		Mid-Range			Complex Site				Digital Workplace	
	Informational	Microsites & Landing	Basic Digital Marketing	Mobile Site	Community-Oriented	Advanced Marketing Portal	Global Enterprise	Multichannel Publishing	Ultra-Large Single	SMB/Departmental	Global/Enterprise
Legacy Platforms											
HP	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
IBM	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
OpenText	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Oracle	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Upper-range Platforms											
Adobe	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
CoreMedia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Percussion	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
SDL	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sitecore	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Putting Scenarios and Vendors Together	Simpler Site		Mid-Range			Complex Site				Digital Workplace	
	Informational	Microsites & Landing	Basic Digital Marketing	Mobile Site	Community-Oriented	Advanced Marketing Portal	Global Enterprise	Multichannel Publishing	Ultra-Large Single	SMB/Departmental	Global/Enterprise
Mid-range Platforms											
Atex Polopoly	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Drupal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
EPiServer	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Hippo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
SharePoint 2013	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Plone	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
TYPO3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mid-range Products											
CrownPeak	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ektron	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Enonic	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
e-Spirit	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
eZ Publish	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Hannon Hill	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ingeniux	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Upland Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Magnolia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
CCI (Escenic)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Putting Scenarios and Vendors Together	Simpler Site		Mid-Range			Complex Site				Digital Workplace	
	Informational	Microsites & Landing	Basic Digital Marketing	Mobile Site	Community-Oriented	Advanced Marketing Portal	Global Enterprise	Multichannel Publishing	Ultra-Large Single	SMB/Departmental	Global/Enterprise
Simpler Products											
DotNetNuke											
Joomla!											
Kentico											
MODX											
OmniUpdate											
OpenCms											
Salesforce											
Telerik											
TERMINALFOUR											
WordPress											

Key	
	Product does not provide this feature in any meaningful way.
	Product provides this feature, but is not as mature as its rivals
	Product provides this feature
	Product excels at this feature, <i>relative to other products in the same category</i>
	Product masters this feature, <i>relative to other products in the same category</i>

How to Use All of these Charts

Scan the scenario summary charts to understand which vendors may best match your business requirements. Look widely across multiple vendors to create a “long list” to investigate. If there are specific functional areas of concern, review the ratings comparisons across specific features. In the product summary charts, review how products’ strengths and weaknesses will stack up against your needs.

For basic technical compatibility, review operating systems, databases, and client data, but when in doubt, always double-check against a vendor’s website, as major software developers are constantly broadening the platforms they support.

If your organization is an RSG subscriber, then you can generate a custom report via our interactive, scenario-based, Custom ShortList Builder™ — available at:

<http://www.realstorygroup.com/ShortlistBuilder/>.

Legacy Platform Vendors: Roll-Up Comparisons

- HP: TeamSite
- IBM: Web Content Manager
- OpenText: Web Experience Management
- Oracle: WebCenter Sites

“Legacy Platform Vendors” are often the oldest players in the WCM market, but with that longevity comes a certain fragility and prevalence of outdated technologies that are becoming harder for these vendors to update and/or maintain. As a result, customers like you are the ones left with various complexities and arduous support efforts.

Over decades of existence, all of these vendors have developed multifunctional Enterprise Content Management “suites”— often via acquisitions of smaller software companies — that address some or even most of the information management lifecycle, from imaging, to DM, to collaboration, WCM, DAM, CRM, and more. On the surface, vendors make a strong case for multifunctional suites:

- Some of their major customers — who have probably already invested substantially in that vendor’s platform and particular API — are clamoring for comprehensive solutions that look at content in a more integrated way
- Obtaining multiple software modules from one vendor should reduce software ownership and support costs and potentially smooth out and accelerate what are always thorny and lengthy integration projects

In practice, the case for ECM suites is substantially less attractive. First, enterprises that already own and have customized systems from other function-point content management software vendors might not wish to replace those existing systems in the name of vendor consolidation. Experienced CIOs recognize that true return on investment comes not just from lower costs, but also from added value to the business lines using the tool. It is critical to remember that the ECM vendors typically possess one or two core competencies, while the packages they acquire have varying provenance. As a practical matter, we do not recommend buying a web content management system from your incumbent imaging or DM vendor or vice versa, simply because you know them.

Additionally, various function point applications within a typical vendor suite remain far from integrated. In the product reviews, you’ll see suites whose individual components often run natively on separate operating systems, support different application servers, and carry divergent repository models. It is telling that ECM vendors use different professional services teams on separate product installations, and are likely to do so for some time.

We have a second and more significant conclusion: web content management now constitutes only a minor part of what most ECM vendors presently offer. Vendors like IBM simply don’t care much about WCM; they make their big money elsewhere. For relatively straightforward web publishing automation, products from the Legacy Platforms tier are probably overkill.

All of these companies are publicly held, so you can scrutinize their financials closely. The “grandfathers” of the WCM marketplace can carry quarter-million dollar licensing price tags to start, moving to seven figures for multi-department, enterprise-wide installations. But the biggest problem here is that customers (and partners) are largely abandoning these systems, with the possible exception of IBM, where their WCM offering is closely tied to their popular WebSphere portal offering (and often tossed in for free). If you license one of these platforms, you are likely joining a shrinking ecosystem.

Specsheet Summary: Legacy Platform Vendors

	HP	IBM
Geography	Global	Global
What's New	<ul style="list-style-type: none"> • Pending excision of Autonomy brand name • Addition of cloud deployment options • Improvements in mobile, social, and campaign management services 	<ul style="list-style-type: none"> • Segmentation and personalization enhancements • Improved WCM social rendering via separate IBM Connections product • Some new Responsive Design themes
Strengths	<ul style="list-style-type: none"> • Broad platform support matrix, and decent hooks into other vendors' commerce and portal servers, including nice preview features • Product focuses intently on marketing-oriented managers' needs • Optional "LiveSite" module helps business users manage dynamic website behavior • Content targeting module is innovative and well elaborated • Company has decent channel of TeamSite-experienced integrators and an unusually vibrant developers' extranet 	<ul style="list-style-type: none"> • Breadth of scalability options, with good guidance around topology • Server runtimes available for a variety of operating systems • Flexible access control and security model, with inherited rights • Localized across (a very impressive) 30+ languages • Excellent taxonomy and metadata facilities could help enterprises with complex categorization and personalization needs • Plays well with other IBM products and services • Experienced and vibrant partner channel, extensive documentation
Weaknesses	<ul style="list-style-type: none"> • A complete solution usually requires licensing multiple products and will complicate your infrastructure, and budget • Underlying TeamSite platform is nearly two decades old; both the architecture and API are unusually heavy and quite byzantine to de-bug • Exceptionally proprietary system requires above-average developer and sysadmin effort • Managing content deployment, synching, and debugging remain complicated and a source of endemic frustration • Company's major bet on IDOL technology for various supplemental services has become a significant drawback • HP will have to completely blow-up the existing platform to align it with modern standards, which means prohibitive upgrade risk for you in the 7.x time frame 	<ul style="list-style-type: none"> • Resource intensive from a hardware perspective • Product's power comes at the cost of formidable complexity and training • IBM WCM is tightly bound to WebSphere Portal • Core WCXM functionality is generally lagging compared to competition • Comparatively meager asset/image management capabilities • User interface is a bit clunky and difficult to customize • Weak in multilingual/globalized content management • Not well suited for granular content reuse • System rollouts tend to be relatively services intensive • Has a poor reputation for quality and performance in publishing (syndication), the availability of APIs, and the ability to integrate with other applications
Potential Fit	Ultra-Large Single Site	Advanced Marketing Portal, Global Enterprise
Unlikely Fit	Advanced Marketing Portal, Multichannel Publishing	All simpler scenarios, Ultra-Large Sites, SMB/Departmental Digital Workplace
Compare To	SDL, OpenText, Adobe, Oracle	OpenText, Adobe, Oracle, SDL
Operating Systems	Solaris, Windows, (prefers) Linux	AIX, IBM OS/400 and Z/OS, Linux, Solaris, Windows
Repository	File system. Supports various RDBMS products for events and reporting	Databases: Derby, DB2 (bundled), Oracle, MS SQL Server

(Continued)	HP	IBM
Client	Browser	Any browser
App Platform	Mixture of Perl, C++, JavaScript, and Java, with occasional legacy Perl; LiveSite runs in standard Java appservers like JBoss	IBM WebSphere Application Server (only)
Licensing	Pricey, with a complex mix of server modules and contributor seats	Based on IBM's "Processor Value Unit" matrix, translating to about \$50K for a single CPU on core Dell PowerEdge Intel XEON processor
Ownership	Public (NASDAQ: HPQ)	Public (NYSE: IBM)

	OpenText	Oracle
Geography	Global	Global
What's New	<ul style="list-style-type: none"> • New inline text, link and image editing feature introduced. • Integration with external content sources like OpenText Content Server, Media Manager, Social Communities, and MS SharePoint 	<ul style="list-style-type: none"> • Merger of some disparate user interfaces into new Contributor UI and Administrative UI • Enhanced mobile web and portal development tools
Strengths	<ul style="list-style-type: none"> • Java APIs and J2EE architecture afford a broad platform support matrix, and a certain degree of developer friendliness • Optional "VBIS" development and runtime environment facilitates integration with a wide variety of remote systems • Wide array of additional modules available for purchase, covering collaboration, rich media services, community content, and more • Potentially very scalable, with a decoupled delivery model and wide array of deployment options for complex network environments • Licensing model is comparatively easy to understand, and bundle pricing is available on popular modules • Developers with OpenText WEM experience are not hard to find • Technology has found a solvent and profitable owner in OpenText 	<ul style="list-style-type: none"> • J2EE-based product that runs on a wide range of operating systems, application servers, and relational databases • A sophisticated development platform for custom content applications • Supports many possible deployment configurations, including tightly, loosely, and decoupled production and delivery environments • Strongly focused on the needs of digital marketers in general and personalization scenarios in particular • Scalability of delivery environment under high loads • Deep LDAP integration can support more centralized security models • Component-based nature of the tool allows for more granular separation of presentation from content, as well as content reuse

(Continued)	OpenText	Oracle
Weaknesses	<ul style="list-style-type: none"> • A potentially very complex product, requiring significant investments in development, support, testing, training, and administration • Somewhat page-oriented system with comparatively weak support for manipulating XML content, and product architecture is somewhat ill suited to component-level content reuse • Java-applet-based rich text editor presents maintenance challenge • Social software applications reside outside the core WCM product • For a platform of this size and maturity, the customer/developer community lacks coherence and visibility • Documentation not par with the product's capability and complexity • Customers report that upgrades can be painful given the company's history of not supporting previous APIs • OpenText is struggling to fit WEM into the rest of its software portfolio 	<ul style="list-style-type: none"> • A definitely very complex system to roll out and maintain • Designing new workflows is an arduous process • Multiple contributor UIs adds complexity and is potentially very confusing for line-of-business users • Dual asset model ("basic," "flex") seems like overkill and adds complexity • Customizing the presentation layer or creating new templates typically requires a Java developer • The caching mechanism is very complex, which may hinder performance in incorrect implementations • FatWire/Sites-experienced developers are not as prevalent as larger competitors are • Potentially very pricey under Oracle's CXM Suite umbrella • Web content management is not Oracle's primary focus — and it shows
Potential Fit	Global Enterprise, Ultra-Large Single Site, Advanced Marketing Portal	Advanced Digital Marketing, Mobile Sites, Global Enterprise
Unlikely Fit	All Basic Scenarios, Global Digital Workplace	All simpler scenarios, most Digital Workplace scenarios
Compare To	HP, Oracle, Adobe, IBM, CoreMedia	CoreMedia, Adobe, OpenText, SDL
Operating Systems	Solaris, Windows, AIX, HPUX, Linux	Solaris, AIX, Windows, HP-UX, Linux
Repository	Database: Oracle, MS SQL Server, DB2	Database: Oracle, MS SQL Server, Sybase, DB2
Client	Browser-based (Firefox, IE and Safari), plus applet for rich-text editor	Browser
App Platform	Java: IBM WebSphere, Oracle (WebLogic, Sun), Tomcat, Red Hat JBoss	Java EE application server: Sun, Oracle, Tomcat, IBM, JBoss
Licensing	Per installation, starting at US\$95,000. Package pricing based on CPU and optional-module combinations — expect to start around \$250,000	Lists at US \$80,000 per CPU for delivery server; other servers and modules are extra
Ownership	Public (NASDAQ:OTEX; TSX:OTC)	Public (NASDAQ: ORCL)

Category Summary: Legacy Platform Vendors

Phase / Attribute	HP	IBM	Open Text	Oracle
Technology				
Technical Administration & Security				
Threat Prevalence				
Authentication & Authorization				
System Reporting				
Multisite Management				
Cloud Services				
Development				
Configuration & Customization				
Integration & Extension				
Content Modeling				
Templating				
Performance				
Back-end Performance				
Site Caching & Delivery				
Content				
Contributor Experience				
Overall Usability				
UI Accessibility				
Contributing Content				
Authoring & Transformation				
Tagging & Taxonomy				
Content Reuse				
Media & Document Management				
Repository Services				
Content Lifecycle				
Workflow				
Globalization				
Archiving & Compliance				
Experience				
Publishing				
Standards Adherence				
Multichannel				
Mobile				

Phase / Attribute	HP	IBM	OpenText	Oracle
Digital Marketing				
Site & Campaign Analytics				
Testing & Optimization				
Segmentation & Personalization				
Social Media Integration				
Promotional Campaigns				
Community & UCG				
Workplace				
Collaboration & Networking				
Dashboard				
Ancillary				
Site Search				
Online Forms				
Module Ecosystem				
Vendor Intangibles				
Vendor Services				
Vendor Professional Services				
Channel Partner Services				
Support & Community				
Strategy & Roadmap				
Viability & Stability				

Key	
	Product does not provide this feature in any meaningful way.
	Product provides this feature, but is not as mature as its rivals
	Product provides this feature
	Product excels at this feature, <i>relative to other products in the same category</i>
	Product masters this feature, <i>relative to other products in the same category</i>

Scenario Fits Summary: Legacy Platform Vendors

Phase / Attribute	HP	IBM	Open Text	Oracle
Simpler Site				
Informational	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Microsites & Landing	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mid-Range				
Basic Digital Marketing	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mobile Site	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Community Oriented	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Complex Site				
Advanced Marketing Portal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Global Enterprise	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Multichannel Publishing	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ultra-Large Single	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Digital Workplace				
SMB/Departmental	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Global/Enterprise	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Key	
<input type="radio"/>	Product does not provide this feature in any meaningful way.
<input type="radio"/>	Product provides this feature, but is not as mature as its rivals
<input type="radio"/>	Product provides this feature
<input type="radio"/>	Product excels at this feature, <i>relative to other products in the same category</i>
<input type="radio"/>	Product masters this feature, <i>relative to other products in the same category</i>

HP: TeamSite

www.autonomy.com

Vendor at a Glance

Specsheet	HP: TeamSite 7.4 Summary
Geography	Global
What's New	<ul style="list-style-type: none"> • Pending excision of Autonomy brand name • Addition of cloud deployment options • Improvements in mobile, social, and campaign management services
Strengths	<ul style="list-style-type: none"> • Broad platform support matrix, and decent hooks into other vendors' commerce and portal servers, including nice preview features • Product focuses intently on marketing-oriented managers' needs • Optional "LiveSite" module helps business users manage dynamic website behavior • Content targeting module is innovative and well elaborated • Company has decent channel of TeamSite-experienced integrators and an unusually vibrant developers' extranet
Weaknesses	<ul style="list-style-type: none"> • A complete solution usually requires licensing multiple products and will complicate your infrastructure, and budget • Underlying TeamSite platform is nearly two decades old; both the architecture and API are unusually heavy and quite byzantine to debug • Exceptionally proprietary system requires above-average developer and sysadmin effort • Managing content deployment, synching, and debugging remain complicated and a source of endemic frustration • Company's major bet on IDOL technology for various supplemental services has become a significant drawback • HP will have to completely blow-up the existing platform to align it with modern standards, which means prohibitive upgrade risk for you in the 7.x time frame
Potential Fit	Ultra-Large Single Site
Unlikely Fit	Advanced Marketing Portal, Multichannel Publishing
Compare To	SDL, OpenText, Adobe, Oracle
OS	Solaris, Windows, (prefers) Linux
Repository	File system. Supports various RDBMS products for events and reporting
Client	Browser
App Platform	Mixture of Perl, C++, JavaScript, and Java, with occasional legacy Perl; LiveSite runs in standard Java appservers like JBoss
Licensing	Pricey, with a complex mix of server modules and contributor seats
Ownership	Public (NASDAQ: HPQ)

Summary

Like its longtime competitor OpenText WEM (former Vignette), Interwoven’s TeamSite Web CMS product grew to become a venerable player in the web content management space — only to be acquired as part of a larger rollup under Autonomy. Then Autonomy in turn was acquired by HP in October 2011 for a whopping \$10.3 billion, a move that degenerated into significant controversy, as HP’s board replaced the CEO shortly thereafter (although the deal still went through).

Subsequent claims by HP regarding “accounting improprieties” by some members of Autonomy’s previous leadership have only added more uncertainty. With no Autonomy leadership left at HP, the vendor claims renewed interest in engineering and a focus on customer support, but as of late 2014, these claims remain largely theoretical.

TeamSite CMS’ history dates back to the mid- 1990s. Interwoven differed early on with emphasis on marketing-oriented customers, developing slick editorial and website management features atop a relatively simple, file-based repository. However, even after a slew of its own acquisitions, Interwoven could not keep up with its larger competitors, and in March 2009, search vendor Autonomy completed its acquisition.

Neither Autonomy nor HP ever upgraded the technical underpinnings of the TeamSite/ LiveSite platform that had been running for nearly two decades. As a result, customers found it increasingly impossible to align TeamSite’s architecture with contemporary standards and performance expectations. In addition, you may need multiple packages for a complete solution, whereas simpler tools may bundle diverse services into one product. Thus, you are bound to deal with an installation that is very complex and difficult to extend and maintain without major engineering undertakings.

The vendor asserts that the product has gone through a number of architectural enhancements, including integration with IDOL and its use as an inverted index (similar to a NoSQL database), as well as improvements in content targeting architecture, page and content templating, and LiveSite Content Services. The reality is that most of these are adjustments around the edges. The core platform remains a developer-intensive conglomeration of proprietary code and methods that only a limited number of people belonging to an esoteric TeamSite guild fully could understand.

Consider TeamSite for applications that involve placing a CMS behind a sophisticated delivery platform, such as a portal or a customer-service knowledge base. It is also plausible for large, static websites — but only those where site managers have higher-than- average-configuration skills, and developers have greater-than-average time and patience.

Scenario Fits	
Simpler Site	
Informational	<input type="radio"/>
Microsites & Landing	<input type="radio"/>
Mid-Range	
Basic Digital Marketing	<input type="radio"/>
Mobile Site	<input checked="" type="radio"/>
Community Oriented	<input type="radio"/>
Complex Site	
Advanced Marketing Portal	<input type="radio"/>
Global Enterprise	<input checked="" type="radio"/>
Multichannel Publishing	<input type="radio"/>
Ultra-Large Single	<input checked="" type="radio"/>
Digital Workplace	
SMB/Departmental	<input type="radio"/>
Global/Enterprise	<input type="radio"/>

The product is overkill for nearly all other scenarios, and is not very well suited for intranets and multichannel publishing scenarios — even if HP can identify some customers who satisfy both of these scenarios.

In the end, we never, ever encounter happy TeamSite customers. You have no compelling reason to join their ranks.

Introduction

Interwoven launched TeamSite in 1995 and like its direct competitors, the vendor built or acquired a bevy of tools targeted at various technology segments across the broader enterprise content management marketplace. Interwoven expanded into new, adjacent areas like asset, document, and configuration management, as well as collaboration and records management. All of these products went to Autonomy in its January 2009 acquisition of Interwoven, and thence to HP in 2011.

In addition to the core TeamSite CMS product, HP offers a variety of corollary products and optional modules. For many installations, several of them will appear quite essential.

- **IDOL Suite:** TeamSite ships with basic repository search; if you need extras, you'll turn to HP's broad catalog of add-on IDOL modules
- **OpenDeploy:** A code and content promotion tool. We have never seen TeamSite implemented without this.
- **MediaBin:** A relatively well-established, web-focused DAM product, added through acquisition, but now part of the DAM brand within HP
- **LiveSite:** Consists of a dynamic delivery module that is explored in-depth under "Experience" on page 169, along with personalization framework LiveSite Targeting
- **Mobile Module:** Adaptive, design-oriented tagging and emulation server, OEM'd from Trilibis

Some products can be combined nicely (particularly TeamSite and MediaBin), but you should understand that, among other differences, several of them have widely varying technical specifications. For example, some modules run inside a Java application server, while others, like TeamSite, have components that are freestanding servers on their own. MediaBin runs only on Windows and customizing it requires ASP/.NET skills. It's all very messy as a group. Other parts of TeamSite are not organic and are based on acquired products, such as the website testing and "optimization" service from former Optimost.

Version 7.4 was released in 2013. HP focused on usability improvements in the CMS by adding persona-based UIs and the iPad app that provides minimal content management on the go for the increasingly mobile workforce in modern world. Cloud-based deployments are now also available via subscription-based licensing.

A dot release (7.4.1) and subsequent fix pack (7.4.1.1) came out in Q1 2014, mostly to address bugs in what was a relatively significant minor release the previous year.

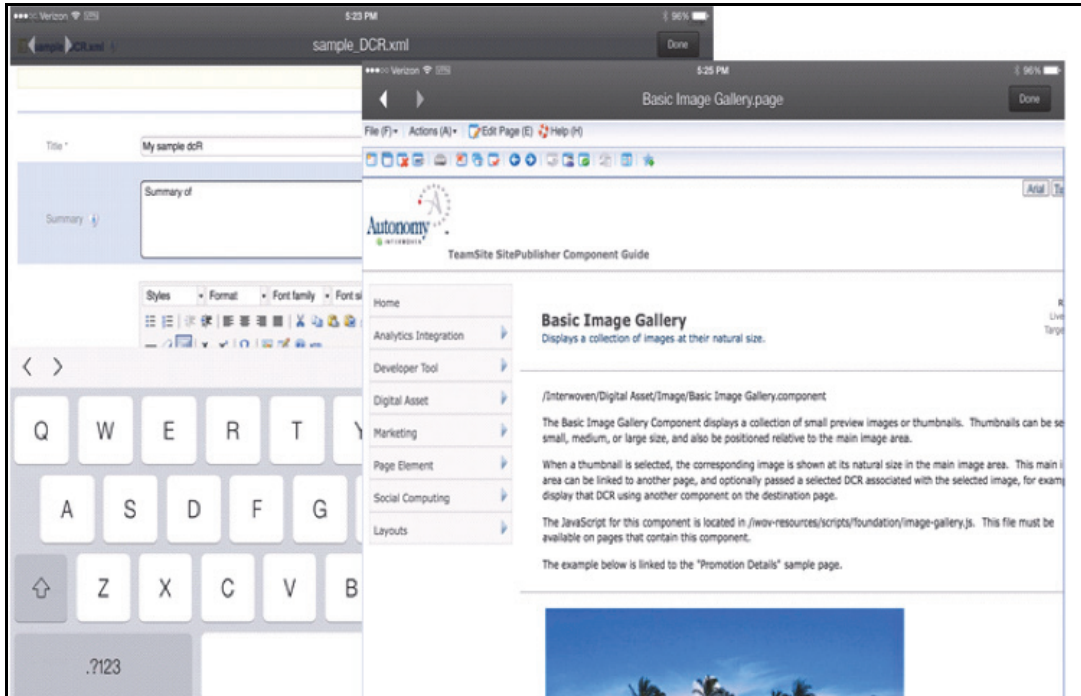


Figure 41. HP TeamSite’s iPad app provides some basic web publishing on the go.

Technology

Technical Administration & Security

TeamSite boasts an impressively broad support matrix. It runs on Windows, Linux, and Solaris, along with many different flavors of relational databases. However, in recent versions, HP has come to encourage Linux, and customers complain that Windows is no longer supported as well as before.

To install TeamSite, you’ll need to run the installer, which is decoupled from the actual packages (.iwpkg). This decoupling approach decreases the overall file size, and makes it easier to release service packs and patches. Note that while the Java-based components can run on nearly any J2EE appserver that supports Beans, the core TeamSite engine, written in C, can only run on simpler, 32-bit hardware.

Compared to most other products, TeamSite is not simple to install. Under the hood, you will find neither new nor standards-oriented technology, and given multiple products and configurations, the typical installation of all the various modules in a production environment is measured in days — not hours.

Technology

Technical Administration & Security

Threat Prevalence	●
Authentication & Authorization	◐
System Reporting	◐
Multisite Management	◐
Cloud Services	◐
Development	
Configuration & Customization	◐
Integration & Extension	◐
Content Modeling	◐
Templating	◐
Performance	
Back-end Performance	◐
Site Caching & Delivery	◐

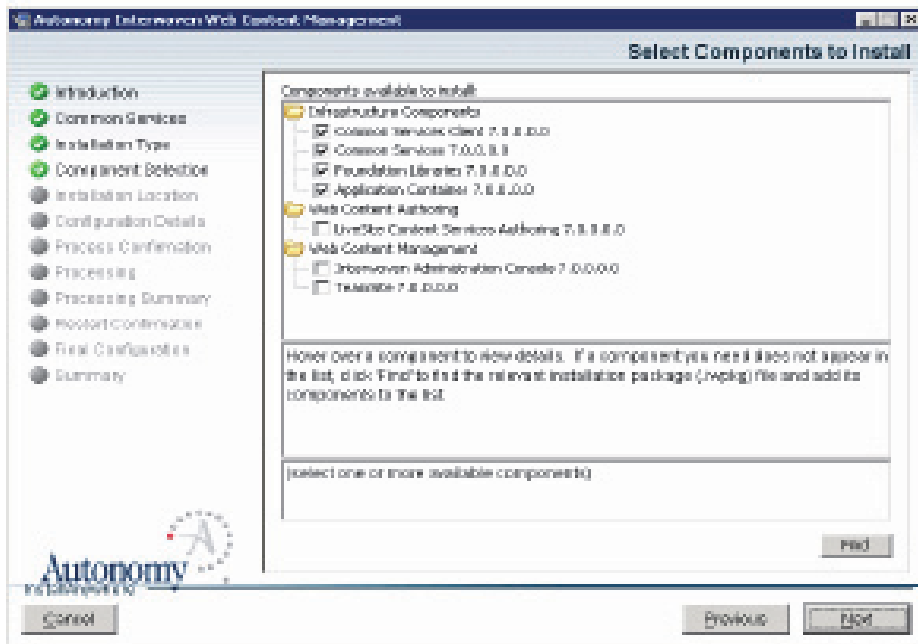


Figure 42. The installation is based on an installer framework that makes it easier to update new service packs and applications.

When you install TeamSite, a typical configuration consists of the following components:

1. **Development Server:** This is essentially the place where new applications and websites are created. It consists of the following:
 - TeamSite with SitePublisher: SitePublisher is bundled with TeamSite and provides a visual interface for creating sites, navigation, adding components and so on
 - OpenDeploy: This is the base server for OpenDeploy, which is used to copy code and content from development server to the runtime server
 - Administration Console: The administration console, which lets you administer TeamSite as well as OpenDeploy
 - Other components that are installed are Application Server (JBoss), Interwoven Registry, and the SDK
2. **Runtime:** This consists of the following components:
 - LiveSite Display Services (LSDS): This is for delivery of content from TeamSite/ SitePublisher
 - LiveSite Content Services (LSCS): The runtime version of LSCS
 - OpenDeploy: The receiver component of OpenDeploy
 - Application Server: Tomcat is used
 - Repository: Based on IDOL search index

As you can see, there are many modules and services. Monitoring all of them becomes extremely complex. An experienced architect told us that, “Installation is a nightmare, with

too many services to monitor. I was tired of monitoring the log files when something didn't work.”

Note that you will probably end up with quite a few databases and schemas with components like LiveSite, TeamSite, reporting, and others expecting their own database.

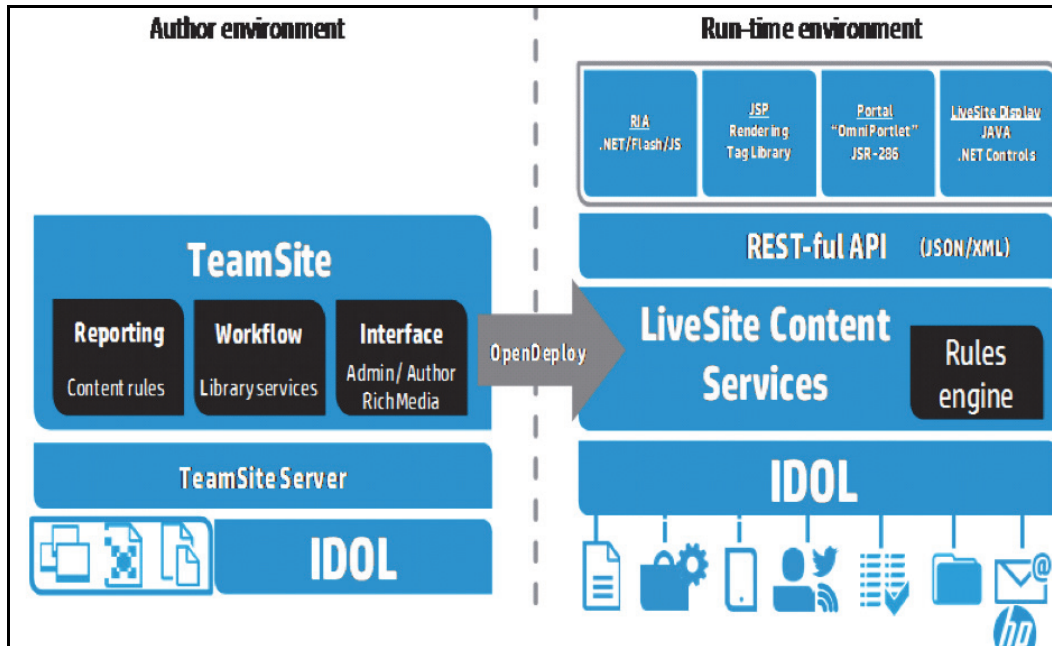


Figure 43. Typical TeamSite architecture separates runtime from the authoring environment. Source: HP.

The vendor positions TeamSite's close neighbor, LiveSite, is an optional content delivery and interaction product — a kind of portal unto itself.

If you don't intend to use LiveSite in your webserver tier, TeamSite can still support "virtualization" against a variety of dynamic delivery approaches, such as major commercial portal packages. It can be tricky to set up, but it works. The product accesses a separate webserver of your choice, where you load your JSP/ASPX/PHP interpreter, and TeamSite loads the presentation template into its rendering engine to give a complete preview for authors in decoupled environments. This has enabled highly distributed enterprises to control some aspects of content management separately, while allowing individual units to run their own website environments.

TeamSite does not use a database to store content (except in the delivery tier when you use LiveSite or some other module), but rather, a proprietary file system-based repository. In TeamSite's "virtual file system (VFS)," contributor's Workareas and Editions contain a copy of every file that makes up the site. HP salespeople will readily demo TeamSite's ability to import and start managing your existing HTML files without the need for a database. TeamSite even conveniently recognizes all of your existing "include" statements.

In TeamSite vernacular, websites or parts of websites are known as "branches" or "sub-branches." Each branch has multiple Workareas (where authors create content), a Staging Area, and Editions. Editors seem to like this approach, and branching tends to ease the maintenance of very large sites. However in the past, some customers have found that the

system tended to top out at about 800 folders. HP’s own documentation recommends as a best practice that you limit the number of directories and files under a Workarea (or another directory) to less than a thousand. To be fair, most CMS packages have such ceilings and we recommend that you do your own testing to determine if there are any implications on performance and manageability.

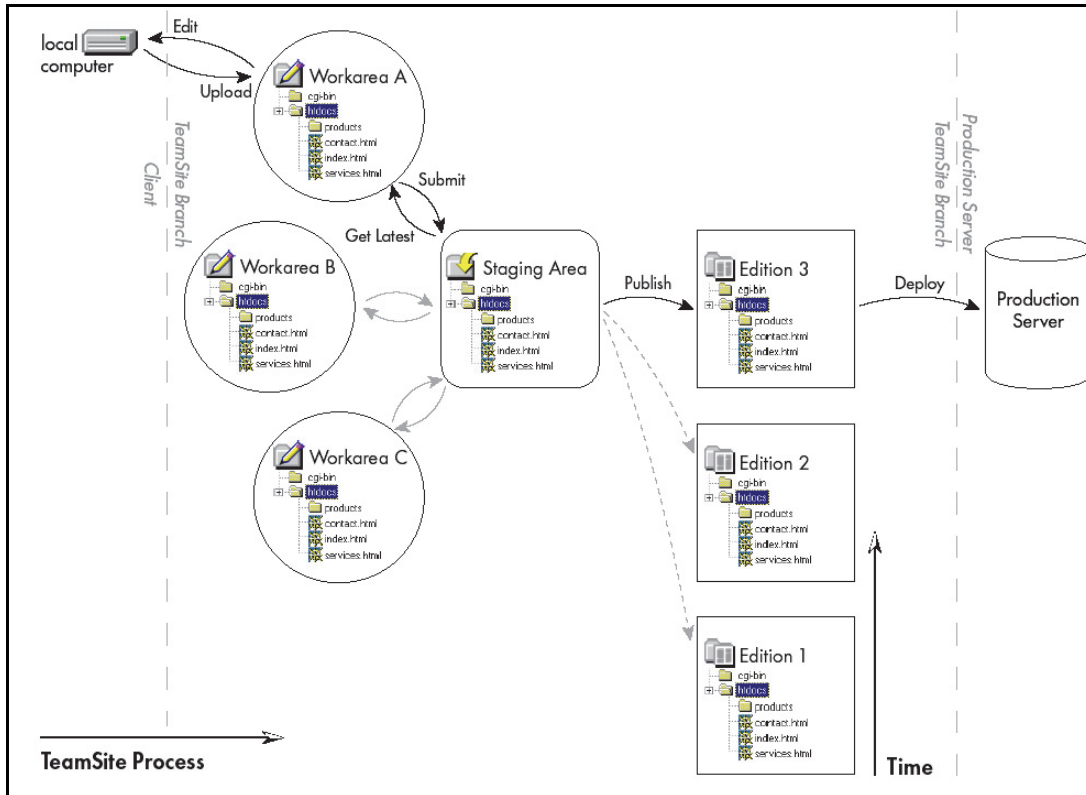


Figure 44. TeamSite has a concept of Branches, which in turn have Workareas, Staging Areas, and Editions.

We think the product’s file system architecture is getting rather old, and it seems to take the company comparatively longer to make system-wide modifications. HP claims that recent overhauls now base the architecture off its IDOL technology. That’s only partially true. IDOL indexes content in the management and delivery tier, but many of the APIs and actions rely on file management and file-based configurations under the content.

In the delivery environment, content does indeed get stored in an IDOL search index, rather than in a database or as XML files. HP claims that this improves performance in write-few / read-many applications, but it is an unusual approach; test to see how it fares with respect to content security and update performance. In addition, it won’t work for any user-generated content, where HP will suggest you apply a relational database. This approach also means that you will have to develop your own interface to get content (such as comments) from a database. HP customers have also told us that IDOL repository corruption happens frequently, which requires re-initialization of the indexes.

You can embed LiveSite tags into other Java applications, which means you can insert TeamSite-managed content into other Java applications without having to use LiveSite as the

principal aggregation platform. Still, you'll find it comparatively more proprietary and non-standard when held up against the other Java-based WCM tools in this report.

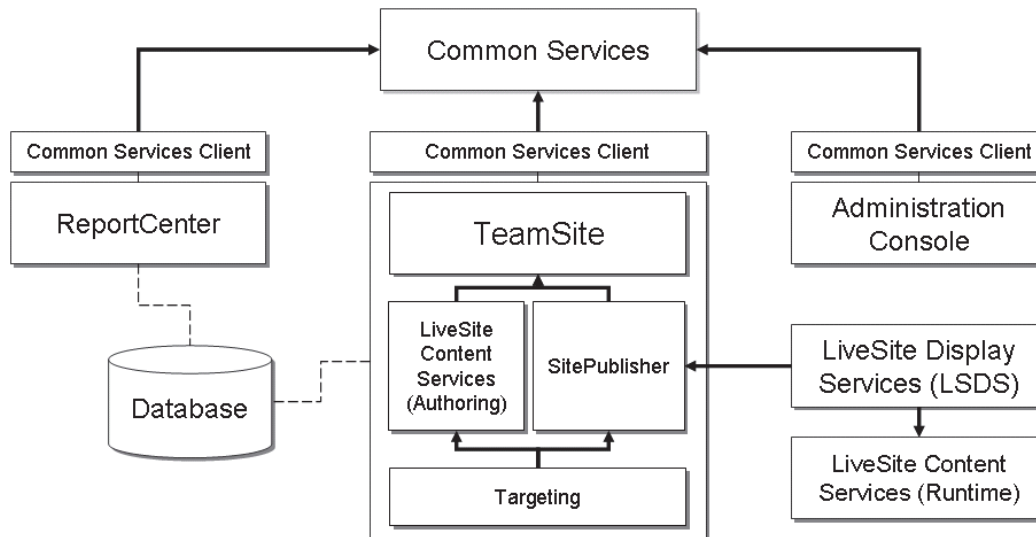


Figure 45. TeamSite building blocks.

To push content — and all other artifacts — from TeamSite to LiveSite or some other delivery environment, you use “OpenDeploy,” a highly sophisticated code and content promotion tool. Like most other tools, OpenDeploy is powerful, with many options and configurations, but customers consistently tell us it is the source of many recurring frustrations. “Painful in all respects,” opined one developer. It is more of a platform than a product, so you will want to keep scripting talent handy.

For publishing and deployment, you have several options:

- Promote HTML files as static pages or snippets to reside on a webserver file system
- Create output templates to “bake” static HTML files by transforming the underlying XML repository
- Transform and push content items into a database (always tricky)
- Transform and push content to the LiveSite repository (an IDOL index)

Deployment of this kind is always tricky, and you’ll want to be sure to lean on an experienced architect here — ideally someone familiar with the sophisticated and sometimes beguiling Interwoven deployment routines. The principal challenges lie in synchronizing different types of content objects: image and other binaries within file systems directories, the content object store, and the underlying metadata repository — which is bound to the content objects on the file system and may need to be mapped to a database. You can avoid mapping by using LiveSite Content Services, which is a NoSQL repository and is schemaless. However, if you want to publish to a relational database, mapping is required.

Similarly, translating schemas to a relational database also takes extra work. The vendor separately distributes pre-packaged integrations with portals. LiveSite portlet can be installed on IBM WebSphere and other JSR 286-compliant portal servers.

In the spirit of modernity, HP launched cloud-based deployment services in its 7.4 release. “Cloud,” in HP’s definition, is managed hosting in the HP cloud infrastructure. What you essentially get is a virtual image from HP’s Professional Services, which will also “massage” it and get it ready with cloud-enabling scripts. Then you can take it to Amazon, for example, and pay HP an annual fee.

The pricing is based on the amount of storage you require and the number of sites and users. HP refers to this as “simplified” pricing, but be careful here; you may pay for growing too fast. There are various Tiers where the medium and larger tiers allow for large storage sizes. Cloud storage may be cheap initially, but can quickly become expensive. This model may be a good fit for you if you plan to launch few new sites, or won’t expand your user base.

Administering TeamSite along with various other modules such as LiveSite and OpenDeploy has traditionally been very difficult, since each product has its own UIs, configuration files, and repositories. The vendor has made significant improvements in this area by introducing a unified administration console where you can manage users, groups, and roles for TeamSite, as well as configurations, handle workflows, and administer deployments for OpenDeploy. Using the same admin console, you can also access logs and reports, as well as configure various connectors.

Remember however, that by using the admin console UI, you can manage only the co-located OpenDeploy base server. If you have other OpenDeploy servers (and you will most certainly have them), you still need to use a separate OpenDeploy admin and user interface.

HP has expended substantial resources on the traditionally weakest leg — TeamSite’s Role and Group Management. Authentication is accomplished through real-time look-up to either an LDAP or Active Directory repository. If an administrator wants to add a new user, she can query that repository first.

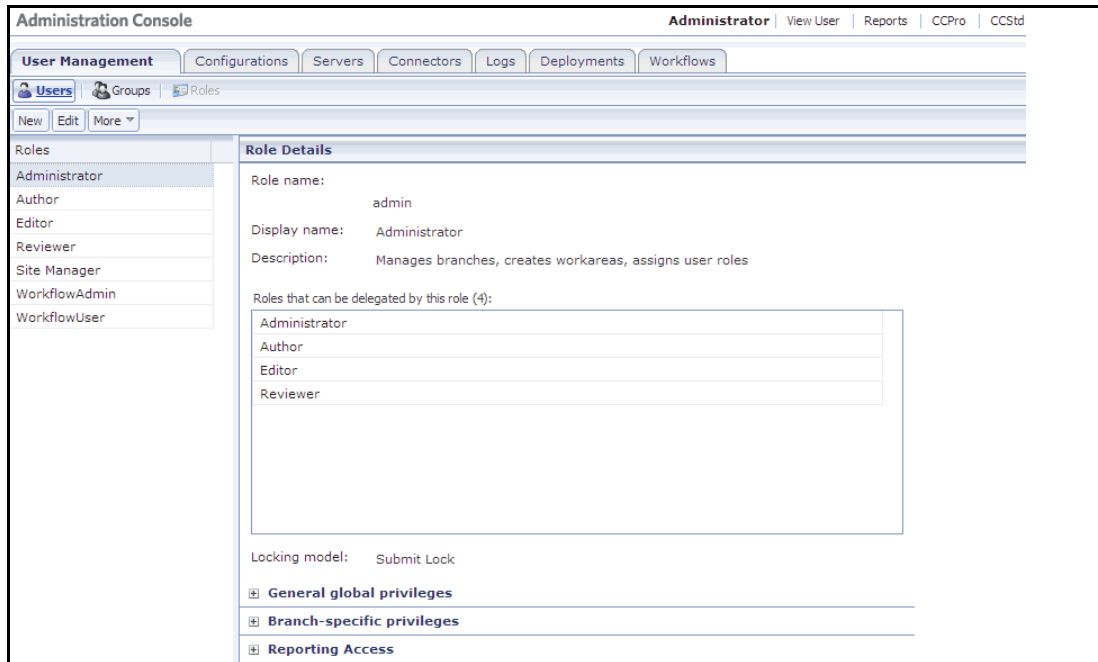


Figure 46. The admin console allows you to manage many configurations directly from the web interface.

Significantly, for the large implementations that HP would like to target, user management can be delegated to local managers. Also, administrators can create custom roles, and users can have different roles across different branches or sub-branches, which is very handy for extranets and microsites.

For reporting, TeamSite offers the ReportCenter as well as OpenDeploy reports.

Development

The most important thing to know here is that TeamSite employs a one-off architecture. One by-product: monitoring and debugging become very difficult, evoking terms like “black magic voodoo” among TeamSite developers. To the extent that the product is driven by a large set of heterogeneous configuration files, understanding what went wrong, and where, can puzzle even the most experienced TeamSite developer.

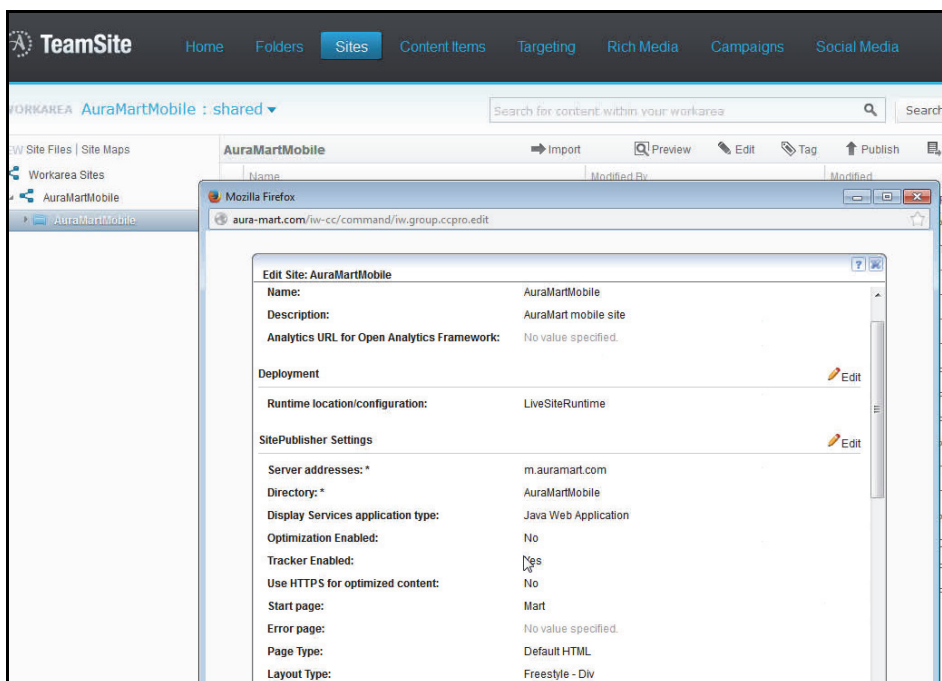


Figure 47. Managing configurations for websites in TeamSite.

You can manage numerous configuration files from within TeamSite, accessing them from the basic admin console. Changes to configuration files are versioned; essentially, you are using TeamSite to manage itself. To be sure, some developers will still prefer to use their own configuration management systems and repositories.

All of TeamSite’s APIs for templating and workflow are JavaScript and XSLT based. (For backward compatibility, the vendor still supports the use of Perl for scripting.) In recent years, Autonomy emphasized XSLT-based templating over a former proprietary approach. While XSLT is more standards-based than some former methods, remember that most other WCM vendors have dropped XSLT as a templating environment over the past decade, even as Autonomy began to transition to it. The reasons for transitioning away from XSLT are well known in the WCM world:

- An unusually steep learning curve and shortage of advanced skills among developers

- A tendency to mix logic and layout
- An inability for business people to make simple changes and other related stresses on your development processes
- Lingering performance concerns

For integration, HP officially counsels using the Java APIs (or alternatively select Web Services wrappers). Some developers have complained that the APIs are buggy as well as impenetrable, and that in particular, it is difficult to access and integrate data content from other repositories at the TeamSite tier. This is not a good content integration platform.

To be fair, with every new release, the APIs are improving and you can actually perform nearly all development without knowing Perl. However, we'd recommend that you still budget for Perl expertise on your team because there are still some areas (like the command-line tools or the ContentCenter interface) and reference files / examples that may require solid Perl knowledge.

Performance

TeamSite has a reputation for running slowly once the number of contributors reaches into the hundreds and beyond, or when content types become reasonably complex. Recent versions have resolved some bottlenecks, but concerns of performance issues continue to be whispered (and occasionally hollered) among major HP customers.

A key shortcoming here is that since TeamSite is not architected as a J2EE or .NET product, your systems engineers will not have recourse to the well-known scaling and performance tuning methods available in those environments. The core TeamSite install essentially runs as a freestanding server (32-bit only). That was the only way to build it sixteen years ago, but it seems quaint now. In any case, you should test and test again, using high volumes of throughput, to make sure the system will handle your loads.

On the delivery side, LiveSite comes with its own caching sub-system; unlike TeamSite, it is based on standard J2EE technologies. The advantage is that you can use well-documented application server tuning techniques and patterns. However, as with any J2EE application, tuning will not be straightforward, so make sure to budget adequate testing time and resources.

At the very minimum, watch caching and publishing performance very carefully, especially if you activate component-based targeting. HP says it can point to major customers using LiveSite in a high-performance environment. Our advice: test performance — and test again — before licensing.

Content

Contributor Experience

The company has condensed its previous multiple and hard-to-navigate user interfaces. Now there is a unified content authoring TeamSite Standard interface, which doesn't look much different from the old one, but it does present a more streamlined and less feature-rich approach. The UI is organized in tabs and is based on roles. HP's hope is that content authors won't be exposed to all existing functionality, rather only to the features that are pertinent to their job roles. In reality (while this move does somewhat simplify the difficult-to-navigate maze that is TeamSite), it doesn't make a considerable difference, according to some users.

However, there's another interface that is available but is not used (or demo'd) as often. It is the pre-existing "Content Center" Professional Edition UI that is oriented more toward power and technical users.

Another shortcoming here is that the various tools in a broader HP suite can sport a plethora of potentially confusing interfaces (like many of its competitors). This is particularly the case with MediaBin, for example.

Content	
Contributor Experience	
Overall Usability	◐
UI Accessibility	◐
Contributing Content	
Authoring & Transformation	◐
Tagging & Taxonomy	◐
Content Reuse	◐
Media & Document Management	◐
Repository Services	◐
Content Lifecycle	
Workflow	◐
Globalization	◐
Archiving & Compliance	◐

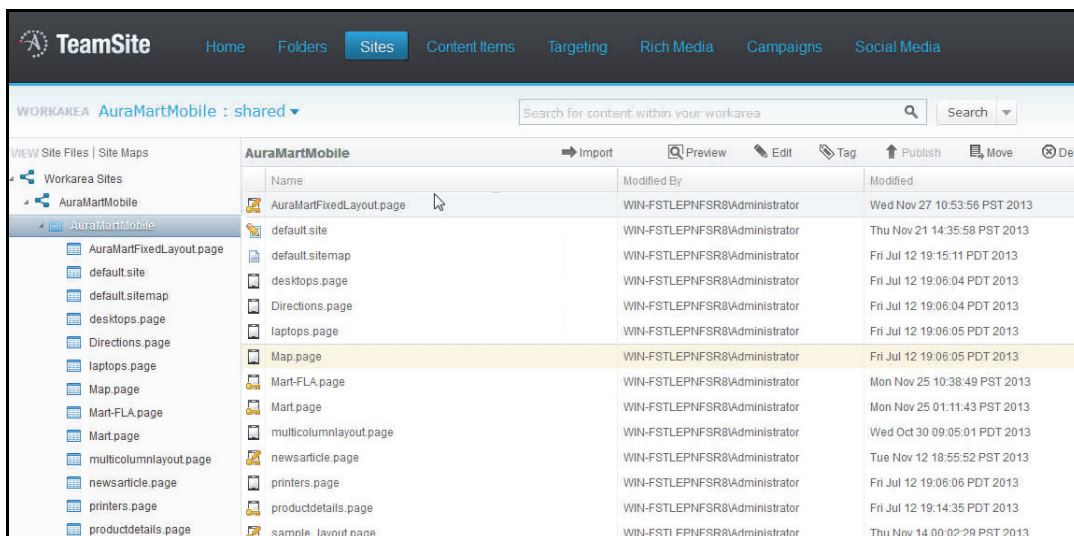


Figure 48. HP TeamSite's user interface.

Both TeamSite interfaces are just frameworks. You can modify them using a nice combination of CSS files and XML configuration files, which are now kept in a separate directory to protect investment upon future upgrades. The config files enable you to hide, change, or add nearly any core functionality from within the CMS. It's not quite a templating system like some other WCM platforms now offer to modify the UI, but it's better than it was before.

needed an extra paragraph. The downside to this approach is that forms with many replicants have traditionally been very slow to load and submit. The company says it has been significantly improved in recent releases; you'll want to test this.

You can save directly to the TeamSite repository from within Word (and use a third-party Word-to-HTML converter), as well as check a document out of WorkSite document management repository and check it into TeamSite, all within Word. There is also an integration with Outlook and TeamSite tasks appear as Outlook tasks. This enables you to handle some workflow tasks from within Outlook. This functionality gets TeamSite a bit closer to satisfying document-heavy intranet scenarios, but note that WorkSite is targeted almost exclusively for the Legal sector.

TeamSite includes the IDOL taxonomy; however, it is difficult to configure it to work well with TeamSite.

Figure 50. Forms-based content entry for creating Data Content Records (DCRs).

Content reuse within TeamSite remains tricky. For content reuse between branches, you can have a component in one branch that consumes the content from a second branch. You can't edit the content within that component, since it should have a single source of truth. However, this approach doesn't make it particularly easy for authors to reuse content outside of their particular "branch," and although TeamSite stores DCRs as native XML, it provides few native tools to assemble and relate content together. Developers have had to create bizarre workarounds to establish relationships among content elements. In general, TeamSite is not a good platform for scenarios involving heavy component content reuse.

The “SitePublisher” module provides an additional interface for content entry as well as for designing websites. You can drag and drop various components into pre-existing layouts or create new layouts. Each component fills a particular slot on the page. To populate a component, you select a particular DCR or define rules for dynamic content based on metadata attributes. Managers can also set personalization rules, access limits, and caching rules at a component level. The SitePublisher module also provides an in-context environment for your editors to edit site content in a wiki-like fashion.

New in recent releases is a somewhat tighter integration with MediaBin and the ability to drag and drop media files from it. Unfortunately — like Adobe and other vendors who sell separate DAM packages — this limits the kind of image manipulation you can do (beyond selecting files) in the CMS itself. For example, to crop images, modify default captions, or create and order slide shows, you have to depart to the separate (also somewhat dated and click-heavy) MediaBin system. Media companies should note that there is no way to distinguish an image collection from an actual slide-show instance.

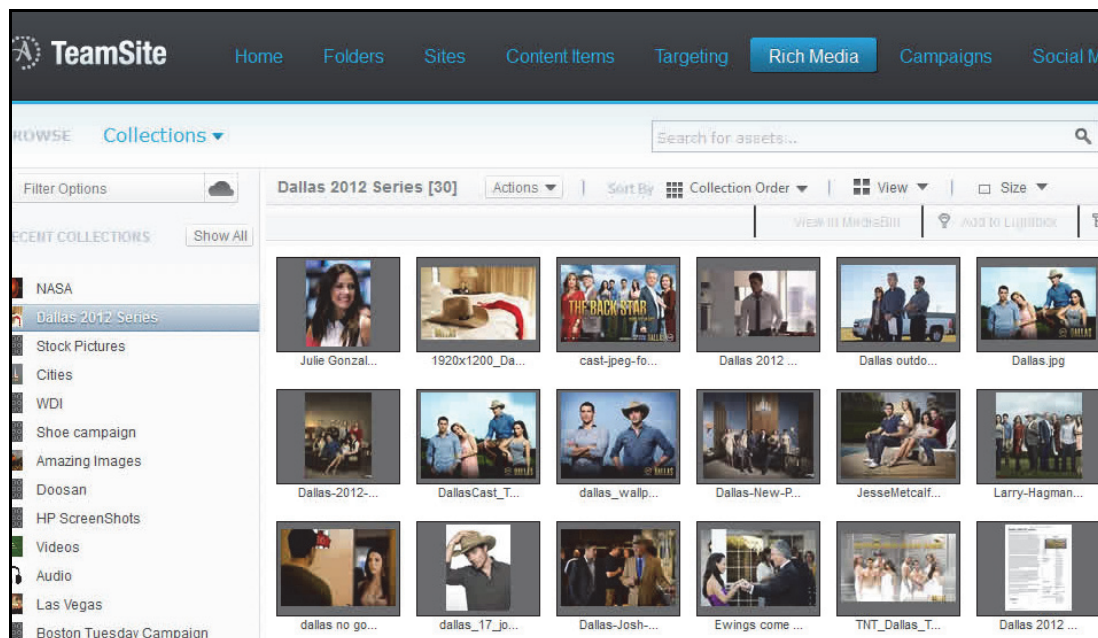


Figure 51. Known as “Rich Media” functionality, this is a hook into MediaBin from TeamSite’s UI.

TeamSite has particularly strong versioning features, allowing for visual reconciliation of branched documents and parallel site development through individual or team work areas. The version control system enables this via several different gradations of content locking (“hard,” “soft,” and “submit”), for different editorial situations that might require multiple contributors working on the same assets.

TeamSite does a good job of handling metadata. It comes with a slick, AJAX-based interface for managing taxonomy nodes in a drag-and-drop environment. However, as with content types, taxonomies are bound to specific branches (think specific websites). This means that you can’t do enterprise-wide web taxonomy management — unless you license additional IDOL modules.

For repository search, you use HP IDOL. It's nicely integrated, and HP will probably stick with it going forward (if only because they paid so much for Autonomy to acquire that technology). In the long run however, it will likely see less innovation than Lucene, which most other WCM packages now use.

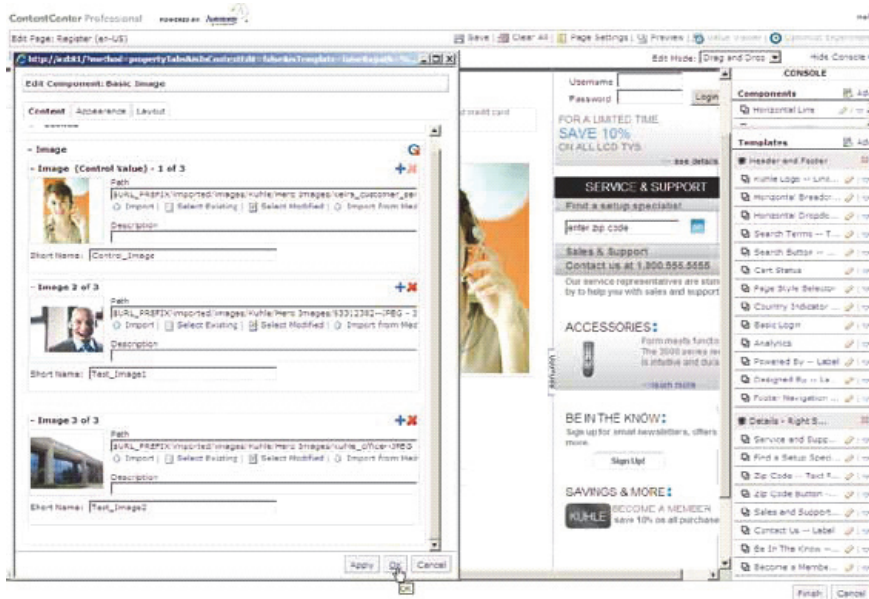


Figure 52. SitePublisher combines inline editing with a drag-and-drop-based page creation interface.

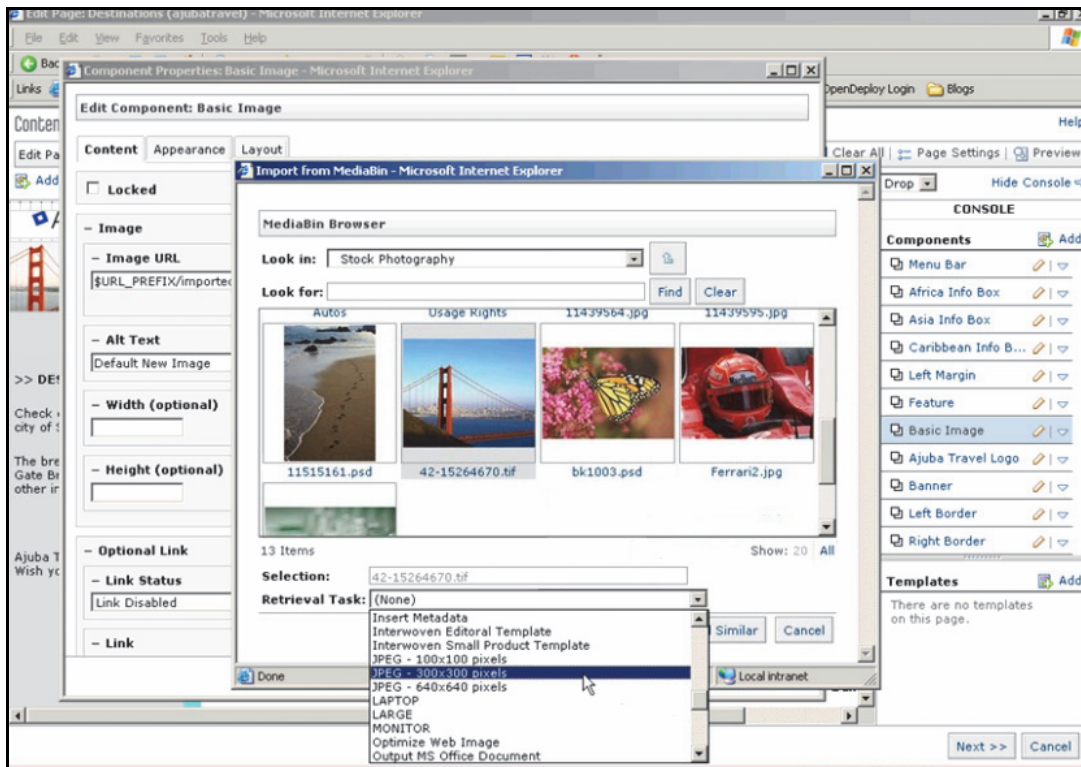


Figure 53. A contributor using a forms-based content entry template in TeamSite can browse or search the MediaBin repository for a high-resolution image that will be automatically copied, down-sampled, scaled as necessary, and transferred to TeamSite when the page is generated and pushed into production. Note the number of pop-ups.

Content Lifecycle

TeamSite’s workflow capabilities are reasonably sophisticated. There is a quite rich task inbox, good notification about jobs, and a useful internal reporting function. The workflow tends to suffer from a surfeit of clicks and excessive jargon (“target node”), which has inhibited adoption for some.

On the plus side, TeamSite had always been strong at task-based workflows, including enabling managers to assign tasks without any content payload — something a surprising number of competitors don’t allow. Editors can annotate content right on the screen using a handy browser plugin (but is not compatible with all browsers).

TeamSite supports “emergency” workflows, which will promote discreet content objects quickly, in lieu of pushing an entire edition (which can take several hours, depending on the size of your site).

Job Properties: 3691 Help

Job Name:

Job Description:

Owner:

Creator: VR1SERVER\Administrator

Priority:

Due Date:

Activated: 8/19/09

Comment	User	Date	Task	
- No Items Available -				
Date	Task	Event	Args	User
Aug 19, 2009 2:16:41 AM	3695	TaskAddFile	b.txt, \default\main\component-guide\testBranch\WORKAREA\testWorkarea	VR1SERVER\Administrator
Aug 19, 2009 2:16:42 AM	3695	TaskActivate	\default\main\component-guide\testBranch\WORKAREA\testWorkarea	VR1SERVER\Administrator
Aug 19, 2009 2:16:42 AM	3695	TaskGoActive	\default\main\component-guide\testBranch\WORKAREA\testWorkarea	VR1SERVER\Administrator

Task	Task Type	Name	Task Description	Owner:
3695	User	Author Work	33	VR1SERVER\Administrator
3707	User	Review 1	Content Approval	VR1SERVER\Administrator
3708	External	End Review Cycle	End Review Cycle	VR1SERVER\Administrator
3710	Submit	Submit	Content Submission	VR1SERVER\Administrator

Figure 54. TeamSite's Workflow.

Email notifications are also unusually rich, and allow editors to access all the dependent pages and assets that may change upon modification of a particular content item. Note, however, that some workflow benefits don't come "out of the box." You should plan for substantial customization here and find developers who have gotten inside the workflow engine before; it is powerful but also complicated and somewhat buggy. "Large workflow projects mean lots of ways to do things and lots of ways to get in trouble," one developer concluded.

TeamSite ships with a web-based workflow modeling tool. The software generates BPMN-compliant XML schemas, which are then translated into TeamSite-specific configuration files upon upload. Those files are stored in the repository where they can be versioned and even workflowed themselves. The modeler works well for structured, consistent workflows, but some users have found themselves back in Perl to implement more complicated conditional logic. The modeler helpfully can invoke OpenDeploy and MediaBin services (assuming you have licensed these tools), as well as call out to remote URLs to pass or receive data and events. On the downside — like much of TeamSite — it seems unusually buggy.

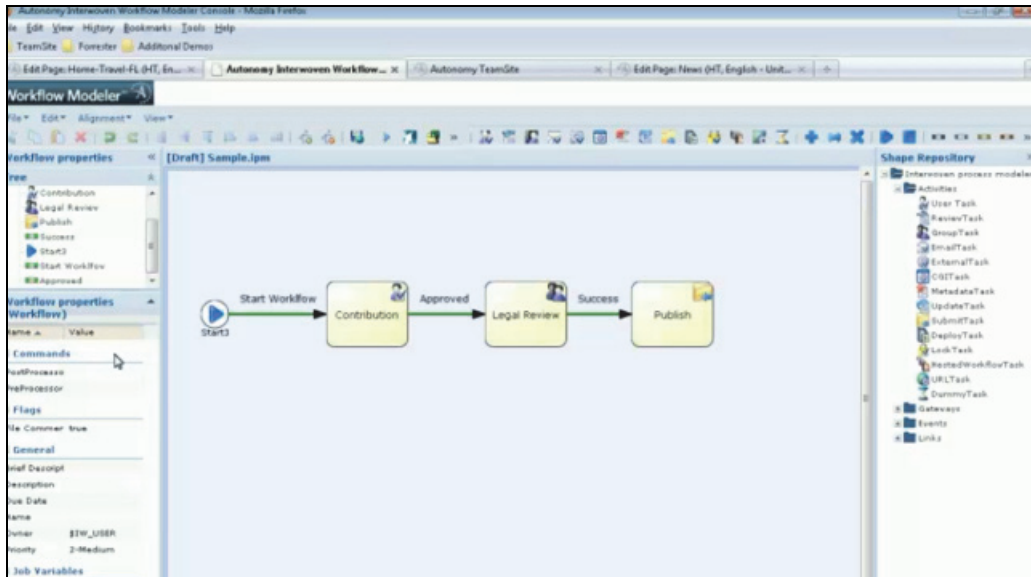


Figure 55. TeamSite offers a web-based workflow designer for creating complex workflows. This is in addition to a desktop version.

Experience

Publishing

By default, LiveSite generates ugly, non-standard URLs. Out of the box, LiveSite URLs look like this:

http://acme.company.com/sites/guide.page?variable=something

However, they can be “massaged” into proper company/product/description patterns via a vanity-URL feature that allows an administrator to set up a particular directory (e.g., /campaignXYZ/) for marketing or advertising tracking purposes (Figure 53). Note that this is different from a more broad-based URL rewriting system for friendly or semantic URLs.

Each page needing redirection must be set up separately. There is presently no programmatic way to generate friendly URLs from existing metadata — nor a manual approach built into the editorial process to name pages — like you would find in almost any other, modern Web CMS package.

There’s one somewhat redeeming feature to mention: if you license the separate Mobility Module (additional fee), the mobile preview is quite good. It ships with a standard emulator,

Experience	
Publishing	
Standards Adherence	<input type="checkbox"/>
Multichannel	<input type="checkbox"/>
Mobile	<input type="checkbox"/>
Digital Marketing	
Site & Campaign Analytics	<input type="checkbox"/>
Testing & Optimization	<input type="checkbox"/>
Segmentation & Personalization	<input type="checkbox"/>
Social Media Integration	<input type="checkbox"/>
Promotional Campaigns	<input type="checkbox"/>
Community & UGC	<input type="checkbox"/>
Workplace	
Collaboration & Networking	<input type="checkbox"/>
Dashboard	<input type="checkbox"/>
Ancillary	
Site Search	<input type="checkbox"/>
Online Forms	<input type="checkbox"/>
Module Ecosystem	<input type="checkbox"/>

but the screen also includes a QR code where authors can snap from any camera-equipped device for a live preview (which in many ways is better than a simulator).

The Mobility Module (OEM’ed from a vendor called Trilibis), provides tags and device detection to support adaptive design, or what Autonomy sometimes calls “server-side responsive design.” You may need to shell out the extra cash for this, since the TeamSite reference implementations are not responsive by default, and it is quite complex to make them so.

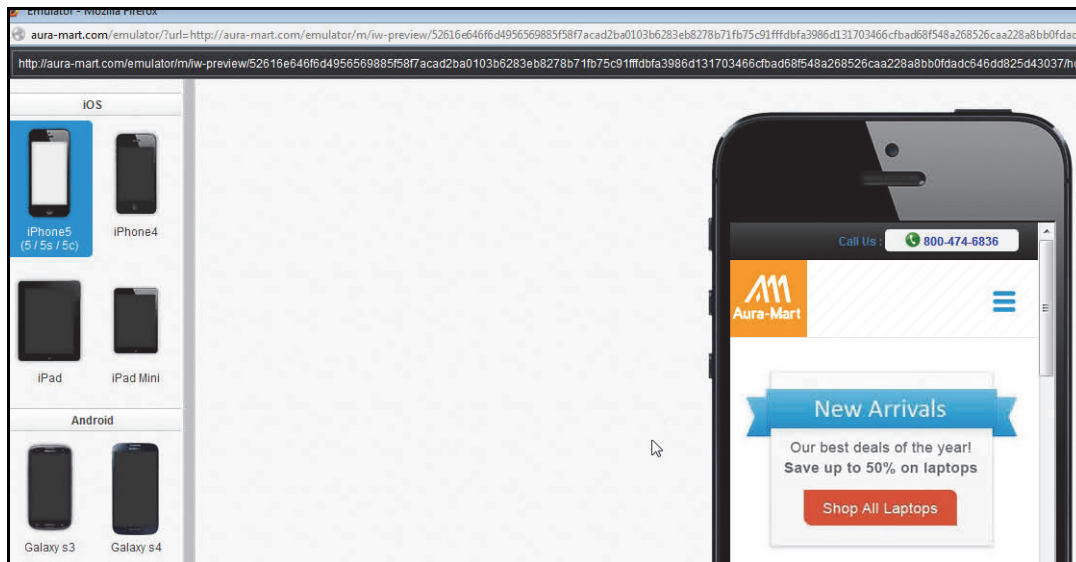


Figure 56. HP uses mobile emulators for iOS and Android devices.

Digital Marketing

For personalization, there’s a separate targeting service that comes bundled with LiveSite. This home-grown rules engine allows you to associate rules to content, based on data (such as customer data from a CRM system, metadata on the page, visitor behavior, and other attributes, such as referrer). The latter represents “anonymous targeting” that allows for things like specific landing pages, based on the search engine keyword queries that sent a visitor to your site — very slick.

Creating rules looked fairly simple to us, but it takes a bit more work — and training — to put them into action. You bind a rule to a “slot” in SitePublisher that has been preset to render according to a rule. Configuration management around reusable logic could present challenges that we’re not sure the vendor fully considered. Also, remember that personalization — or even group segmentation — can be the enemy of caching...and therefore, performance. Test carefully.

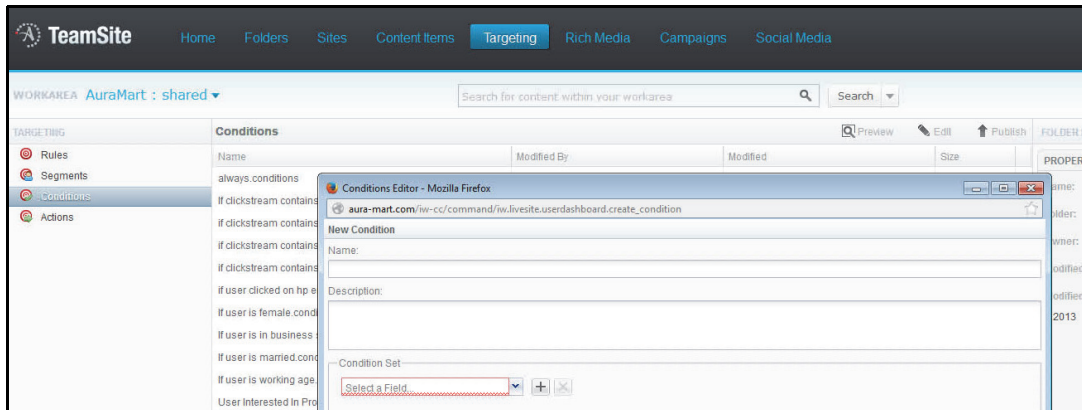


Figure 57. Setting up conditions for targeting rules in HP TeamSite. It is unlikely your marketers will find this task (and interface) easy to use.

With respect to integrated traffic metrics, TeamSite is most closely integrated with Webtrends. If you're a Webtrends licensee, you can drag and drop analytics tags on pages, as well as view metrics reports in context of the web page directly. Unfortunately, you can't do it with Google Analytics or Adobe SiteCatalyst.

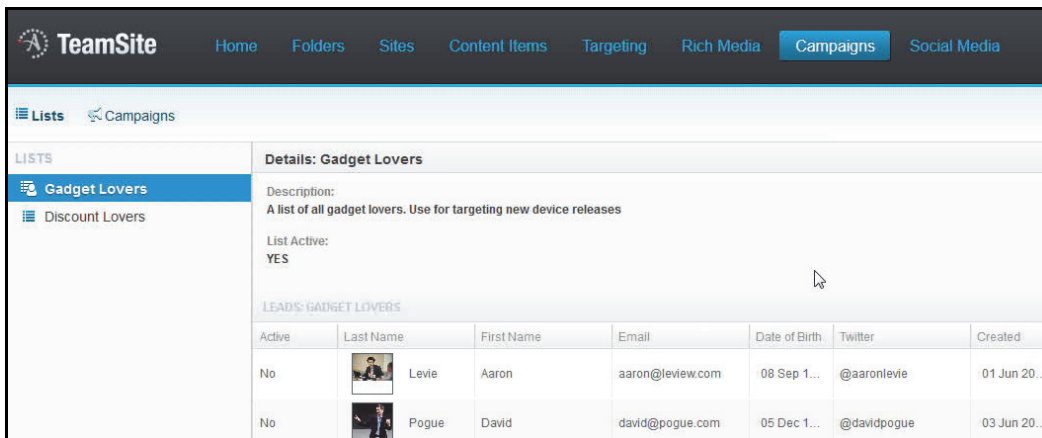


Figure 58. Campaign management tab in TeamSite's UI.

More broadly, TeamSite ships with default page-tag elements for Webtrends and Google Analytics, as well as generic JavaScript containers where you can put other vendors' tracking code. That's the extent of what you can achieve through this "integration."

HP also provides what it called its "Open Analytics Framework." That's a fancy name for some services that are likely to emerge in some future edition.

LiveSite now ships with 100+ starter components, for things like blogs and slide shows. HP offers samples for each, but some customers have found them a bit thin and have asked for more reference implementations to see the code in action. On the other hand, when sufficiently trained, marketing staff at some large customer sites have come to value the fine-grained control over interactivity that these components — and LiveSite more generally — can afford.

In recent releases, the vendor has worked to deepen LiveSite's community and UCG capabilities, primarily through blogging and other social components. Note that all comments

or other user-generated data is written to the delivery database and won't go back to the core content repository in the contribution environment, unless you set up a job to push it there. You can also configure any LiveSite URL to auto-generate an associated RSS or ATOM feed.

Ancillary Services

For site search, you employ Autonomy's IDOL, or another best-of-breed search engine, depending on how you publish your website. More broadly, there is no third-party ecosystem of LiveSite modules, like you often find with other WCM vendors.

Vendor Intangibles

Like the other vendors at this tier, customers have had rather negative impressions of HP/Autonomy's first-line support. Perhaps the product innards are simply too complex for traditional call centers, and in any case, the company's professional services and support teams seem to apply different and varying methodologies to the customization process.

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

Meanwhile, even long-standing partners complain that bug requests disappear into the ether, or that the vendor continued to promise that a fix would come in the next version. On the other hand, some customers have lauded TeamSite's second-line support. More than one customer recounts stories of experienced engineers shipping them custom scripts to "undo" mistakes. Similarly, customers tend to praise the documentation, which is thorough and filled with good code snippets, although it's sometimes difficult to search and parse. Unlike some other vendors, documentation is not freely available to everyone

Count on several months to get your developers fully versed on the product. Ideally, you would be able to find lots of certified, independent specialists out there, but as with its competitors, experienced, TeamSite-savvy engineers are at a premium. Remember that this is a comparatively dated and arcane system — plan accordingly.

On the plus side, HP has reinvigorated Autonomy's lagging user conference. The most recent event in late Q3 2014 attracted 650 participants to six tracks of sessions — one dedicated to TeamSite/LiveSite. However, you will look in vain for local user groups.

When looking for outside help, note that TeamSite does enjoy a fairly broad network of systems integration partners. TeamSite/LiveSite is a platform that demos very well, but both the vendor and its consulting partners tend work very hard to conceal its technical complexity.

As with other vendors at this tier, HP TeamSite is priced according to a mix of user and server licenses, with a slew of optional modules that you may not need — or even use (MediaBin, LiveSite, WorkSite, et al.). LiveSite is licensed based on per transaction metrics, but unlike other TeamSite modules, you install it in your delivery environment as well — which could be numerous servers. This is what Wall Street praises as "obtaining a higher yield" from existing customers. In addition, you have to pay for development and staging licenses (be sure to negotiate hard here), which cannot commingle on the same hardware.

To be sure, it is ultimately the customer's responsibility to weigh the suitability of any deal. As with OpenText (or any other vendor at this tier), you'll want to negotiate very carefully. Ask a lot of questions, and insist on a synchronization of technical and price proposals.

Conclusion

At the time of the Autonomy acquisition, Interwoven enjoyed good visibility within the marketplace. Autonomy never did kill this golden calf, but they didn't seriously deign to upgrade it, either. Consequently, the core platform continued to fall behind amid Autonomy's strange, almost fetishist attention to IDOL integration. Today, the product is oversold, needlessly packaged as a bloated suite, and quite over-hyped amid a marketplace that has surpassed it.

HP points renewed investment on the engineering side, but thus far, we've yet to see this transpire, given the major overhaul this platform desperately needs. The plethora of tools here has many moving parts that require extraordinary attention and maintenance, but HP's avowed future of an Adobe-like Marketing Cloud (really just a dashboard above the suite) doesn't solve its core problems, which run more deeply.

Overall, when you look at TeamSite's usability, architecture, features, implied development models, and overall vendor approach, you feel teleported back to the 1990s. It's a Frankenstein of a platform that only an avowed TeamSite guru could love.

TeamSite salespeople will tell you that the fact that the system has never been significantly modernized is actually a plus, since their customers have never gone through a major re-write. This is inane and false on the face of it, since this developer-heavy — and quite proprietary framework — has left many customers forked from the upgrade path and languishing on an un-dead version of the platform.

Every year we search for a happy — or at least contented — TeamSite customer; we can never find one. The fact that this platform continues to get top-right rankings in nearly every major analyst quadrant speaks more to the attractiveness of TeamSite demos and the gullibility of busy evaluators (not to mention Autonomy's and now HP's very, uh, persuasive analyst relations). More than any other product in this report, TeamSite leads to chronically unhappy business users and developers.

To be fair, HP has proven a responsible enough owner such that if you are an existing TeamSite customer, you need not rush to exit. However, if you are a new prospect, we see no compelling reason to license this platform — all the more so since a long-promised future "Version 8.0" would have to represent a nearly complete rewrite to gain any credibility as a true contender.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

IBM: Web Content Manager

www.ibm.com

Vendor at a Glance

Specsheet	IBM: Web Content Manager 8.0.0.1 Summary
Geography	Global
What's New	<ul style="list-style-type: none"> • Segmentation and personalization enhancements • Improved WCM social rendering via separate IBM Connections product • Some new Responsive Design themes
Strengths	<ul style="list-style-type: none"> • Breadth of scalability options, with good guidance around topology • Server runtimes available for a variety of operating systems • Flexible access control and security model, with inherited rights • Localized across (a very impressive) 30+ languages • Excellent taxonomy and metadata facilities could help enterprises with complex categorization and personalization needs • Plays well with other IBM products and services • Experienced and vibrant partner channel, extensive documentation
Weaknesses	<ul style="list-style-type: none"> • Resource intensive from a hardware perspective • Product's power comes at the cost of formidable complexity and training • IBM WCM is tightly bound to WebSphere Portal • Core WCXM functionality is generally lagging compared to competition • Comparatively meager asset/image management capabilities • User interface is a bit clunky and difficult to customize • Weak in multilingual/globalized content management • Not well suited for granular content reuse • System rollouts tend to be relatively services intensive • Has a poor reputation for quality and performance in publishing (syndication), the availability of APIs, and the ability to integrate with other applications
Potential Fit	Advanced Marketing Portal, Global Enterprise
Unlikely Fit	All simpler scenarios, Ultra-Large Sites, SMB/Departmental Digital Workplace
Compare To	OpenText, Adobe, Oracle, SDL
OS	AIX, IBM OS/400 and Z/OS, Linux, Solaris, Windows
Repository	Databases: Derby, DB2 (bundled), Oracle, MS SQL Server
Client	Any browser
App Platform	IBM WebSphere Application Server (only)
Licensing	Based on IBM's "Processor Value Unit" matrix, translating to about \$50K for a single CPU on core Dell PowerEdge Intel XEON processor
Ownership	Public (NYSE: IBM)

Summary

IBM Web Content Manager (formerly IBM Lotus Web Content Management) is the WCM piece of Big Blue’s WebSphere stack, providing WCXM functionality for WebSphere Portal customers. In theory, WCM is a standalone product that doesn't require the purchase of other Big Blue stackware. In reality, IBM WCM is part of the very complex IBM Customer Experience Suite, is tightly bound to the WebSphere Portal, and requires several other IBM offerings for a complete solution.

IBM released version 8.0.0.1 in January 2013 with the idea of advancing its Web Experience Management capabilities.

Although WCM is replete with useful functionality and highly extensible, no one would accuse it of having an industry-leading feature set or setting the bar for innovation. In fact, on features alone, it is generally lagging behind many of the upper-tier players, particularly in terms of experience management. Without the backing of IBM, it's hard to imagine WCM faring well on its own “in the wild,” against many of the richer, more innovative offerings from other vendors in this research.

Nonetheless, IBM claims more than 10,000 WCM licensees worldwide (the vast majority of which are portal licensees using WCM). Based on anecdotal interactions with customers, it seems that most came via bundled deals driven by IBM's broadly deployed WebSphere stack.

In the end, WCM remains best suited to scenarios in which an enterprise has already made a substantial investment in WebSphere Portal and related solutions. Others should consider their requirements carefully against WCM's capabilities, keeping in mind that despite a reasonable amount of usable capability out of the box, WCM likely will require a significant investment in IBM services to get up and running.

Customers who have already invested in a WebSphere Portal deployment stand to benefit the most from WCM, although such an investment should by no means automatically dictate choosing WCM over other possible contenders. It is unlikely to present a compelling reason for customers outside the WebSphere ecosystem to invest in it.

Scenario Fits	
Simpler Site	
Informational	<input type="radio"/>
Microsites & Landing	<input type="radio"/>
Mid-Range	
Basic Digital Marketing	<input type="radio"/>
Mobile Site	<input type="radio"/>
Community Oriented	<input type="radio"/>
Complex Site	
Advanced Marketing Portal	<input type="radio"/>
Global Enterprise	<input type="radio"/>
Multichannel Publishing	<input type="radio"/>
Ultra-Large Single	<input type="radio"/>
Digital Workplace	
SMB/Departmental	<input type="radio"/>
Global/Enterprise	<input type="radio"/>

Introduction

The product known as IBM Web Content Manager (IBM WCM) has had various monikers over the years. In July 2003, IBM acquired Australian CMS vendor Presence Online, who had rolled out “Apatrix.” Several years earlier, Apatrix was one of the first and most notable CMS products built atop Domino.

IBM rebranded the Apatrix product as “IBM Lotus Workplace Web Content Management,” or ILWCM, (pronounced “Il-Wiccam” in IBM-speak), slotting it under the broader Workplace collaboration rubric. For a short while, it was just WWCM, but the name was changed to Lotus

Web Content Management. That moniker stuck for a while, but yet again, IBM renamed the product to IBM Web Content Manager in March 2011.

The Aptrix product was originally available in Domino and Java versions. After two years selling both products side-by-side, IBM finally decided to sunset the Domino version, in favor of standardizing on Java EE. Thus, despite the Lotus branding, WCM has really been a (strictly Java-based) WebSphere product.

Several releases of IBM WCM focused on integrations with Portal and the ability to expose content via different types of presentation components like widgets and portlets on the same page — all in the name of keeping the IBM stack more or less intact, and to appeal to enterprise buyers who may favor many pieces of software from the same vendor.

Technology

Technical Administration & Security

Architecturally and otherwise, the most significant aspect of this product is its deep integration with WebSphere Portal Server and its use of the Portal infrastructure as its runtime framework. You access WCM's content-production facilities through WebSphere portlets, and — by default — content is published out via the Portal. Note that if you buy WebSphere Portal, you'll get IBM Web Content Manager with it. Conversely, if you buy WCM, you get the WebSphere Portal runtime license along with it, but you're restricted to use it only for creating, maintaining, and delivering web content. No matter how you obtain WCM, the package comes bundled with IBM Tivoli Directory Server, Apache Derby, WebSphere Application Server Network Deployment (not the full edition, but enough to run WCM), and portions of Lotus Quickr. Already, that's a fair amount of latent functionality — a formidable number of moving parts for you to manage and master, and a significant performance overhead.

You can optionally use IBM Content Manager (the company's long-standing document and asset management solution) as the core repository, a choice that makes a certain amount of sense for files and unstructured content. This means, among other things, that you can take advantage of Content Manager's famously extensive library services and content federation capabilities. The FileNet ECM suite is yet another repository option. Both of these options — along with WCM itself — support the CMIS standard for content interoperability, which makes it possible to connect WCM with repositories of some enterprise content management products, but remember that this effort is not a push-button affair. The payoff for jumping through the CMIS hoop however is that accessing both content and access control listings will be more straightforward than before. The portal has further integrations with SAP, allowing the population of portlets with content residing on NetWeaver portal (including iViews and portal pages — indeed anything that is URL accessible). For intranet scenarios, this could be useful.

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

Traditionally, WCM has not been as mature technically as most of IBM's other Java-based tools. It has had a reputation among developers for being buggy, although some of this stems from licensees not keeping up with the frequent product patches. (Remember, you're looking at patches involving WebSphere, WebSphere Portal Server, and other infrastructure pieces in addition to WCM itself.) Because of the many touchpoints between the underlying core pieces, you shouldn't be surprised to find that the potential for unexpected upgrade- and fixpack-related glitches is large. Again, this is something customers and consultants frequently mention as a source of frustration. IBM says this has been a focus area for them and they have invested a lot into making this easier for customers, resulting in the version 8.0 release. In any event, plan to follow the IBM best-practice guides very closely here. Fortunately (unlike many of its competitors), IBM does provide a wealth of setup, tuning, capacity planning, and architectural best practices info for its various products — including WCM.

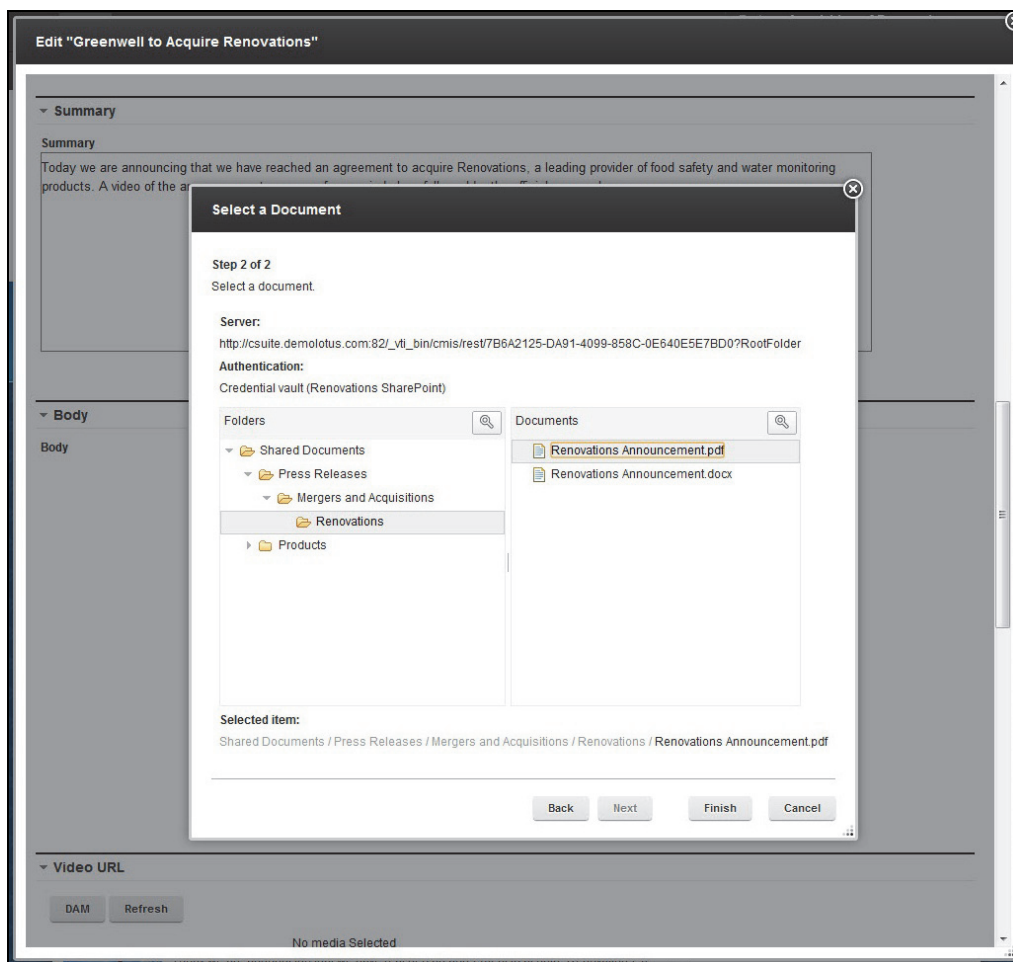


Figure 59. Importing content via CMIS into IBM WCM from SharePoint.

Overall, the product seems well architected (following well-accepted Java EE patterns) and rests on mature foundations. WebSphere is well proven to be durable in practice.

With respect to publishing, the system is flexible. Out of the box, in single-instance mode, WCM is a non-tiered, fully coupled product in which all management and delivery runtimes

run on the same physical box (operating behind the firewall). More typically, though, the system is deployed in multi-tiered fashion and content is pushed out to “subscriber” machines by a syndicator machine. Syndication can be daisy-chained across multiple delivery machines so that content can be “fanned out” by syndicators that supply other syndicators (to avoid all the load falling on just one syndicator machine). Note that each subscriber machine needs a licensed instance of WCM in addition to each syndicator. License costs can mount quickly.

Syndication is not without its drawbacks. Developers have told us it's difficult to customize syndication behavior (such as creating different schedules for each syndicator or programmatically accessing syndication functionality via APIs).

Promoting business logic and other CMS components from a development to production environment can be more challenging than syndicating content, particularly because the system mitigates against using multiple development instances. However, it depends on what kinds of artifacts you're trying to deploy; in recent versions, the promotion process has been unified to a degree — allowing portlets, access controls, and content to be deployed as a single package

WCM comes with a baking routine to generate content for broad website delivery. Like its competitors' offerings here, IBM's pre-generation system has been finicky and unpredictable. Some licensees have resorted to less frequent (e.g., once-a-day) editions to reduce resource loads. It's up to you to promote the files to your production web server tier; the product offers no native deployment services here.

The system allows you to set publish and expire dates on content, such that if a content item is eligible (by virtue of its date) for publication, it goes into the syndication queue automatically. Conversely, an item that has reached its expiration date is unpublished automatically.

The product employs WebSphere's “member manager” functionality for security (including authentication and single sign-on), and — not unexpectedly — WCM integrates well with the Tivoli identity-management stack. User management at the administrative level is relatively strong; you can lock down content at any level in the hierarchy, down to content elements, and there is ample support for roles. Role privileges are inheritable. (This is a rare feature in the CMS world, but few customers really need it.) CMIS support adds the possibility of utilizing access control (via formal ACL) inherited from other similarly compliant repositories. You won't long for power or flexibility in managing users, groups, roles, and rights in WCM. If anything, there may be too much power for many scenarios. The system makes it easy to create overly complex entitlement relationships. Use caution.

IBM WCM also supports OpenSocial integration with Google, Facebook, and Yahoo!, as well as OpenID, which allows user authentication via the same services.

Except for syndication and publication events, WCM isn't overly strong in “at-the-glass” report rendering, even though it provides good capabilities for data collection.

To monitor the health and performance of your setup, you access the application server administration console via its Performance Monitoring Infrastructure. Using this, you can track parameters at the component level and find out if there are bottlenecks. The other alternative is to buy a separate monitoring package like Tivoli Composite Application Manager.

Development

In demos, IBM will tout its RAD development environment called Web Experience Factory (formerly WebSphere Portlet Factory), which came to IBM from its BowStreet acquisition. Mainly, it shields the developer from the underlying APIs and comes with 125 wizard-like builders for different types of portlets, including popular integration points. In Web Experience Factory, you can deploy an application across multiple channels like most popular smartphones and tablets – either iOS or Android — in addition to targeting IBM WebSphere Portal, IBM Mashup Center, and IBM WebSphere Application Server). However, experienced developers tend to think of Experience Factory as a demo/prototyping environment only (akin to “SharePoint Designer” in Microsoft), and often don’t trust the code it generates. More importantly, once you touch the code itself, the Factory becomes useless for you going forward as some users have commented.

More seasoned developers — including IBM’s own professional services teams — will tend to use Rational Application Developer without using the Web Experience Factory, to work directly with the Java code.

Note that although WCM does come with a templated sample site out of the box, much of what you’ll see in a sales demo requires some developer work to set up. This is true of most products at this level, but it’s especially true for WCM. Make a point of asking how the specific pieces you’re seeing demonstrated were set up; don’t assume they came that way out of the box.

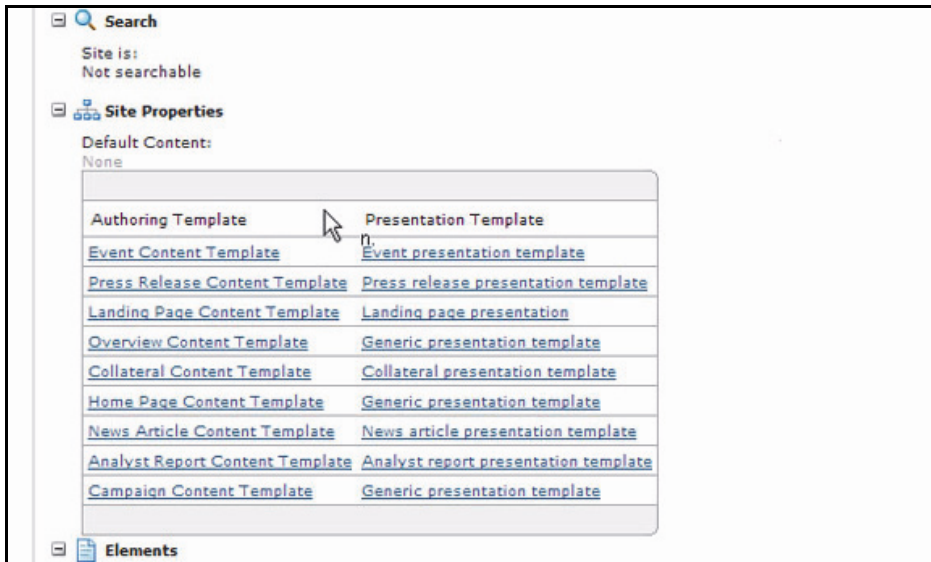


Figure 60. Administrators map content types to display templates. In general, the product is better suited to content repurposing than it is for highly granular content reuse.

In WCM, content types can be created as freestanding content components (“elements”), or as “authoring templates.” You can conveniently develop them in a browser, and then you bind them to “presentation templates,” or reference them directly from a site page.

Like content types, “presentation templates” can be created in forms as well, although some components may need to be added manually using the WCM tag library. You also can use

Dreamweaver with a special plugin that exposes the WCM taglib for layout design. Like HP, OpenText, and others, IBM comes with some pre-set page components, such as a tree-navigator menu, or components for inserting canned queries. The bottom line is that you control delivery behavior from the same administrative interface used for managing the CMS, which certainly makes things simpler. IBM has this luxury, because it has embedded its own delivery environment in the CMS. This approach is clearly not for everyone.

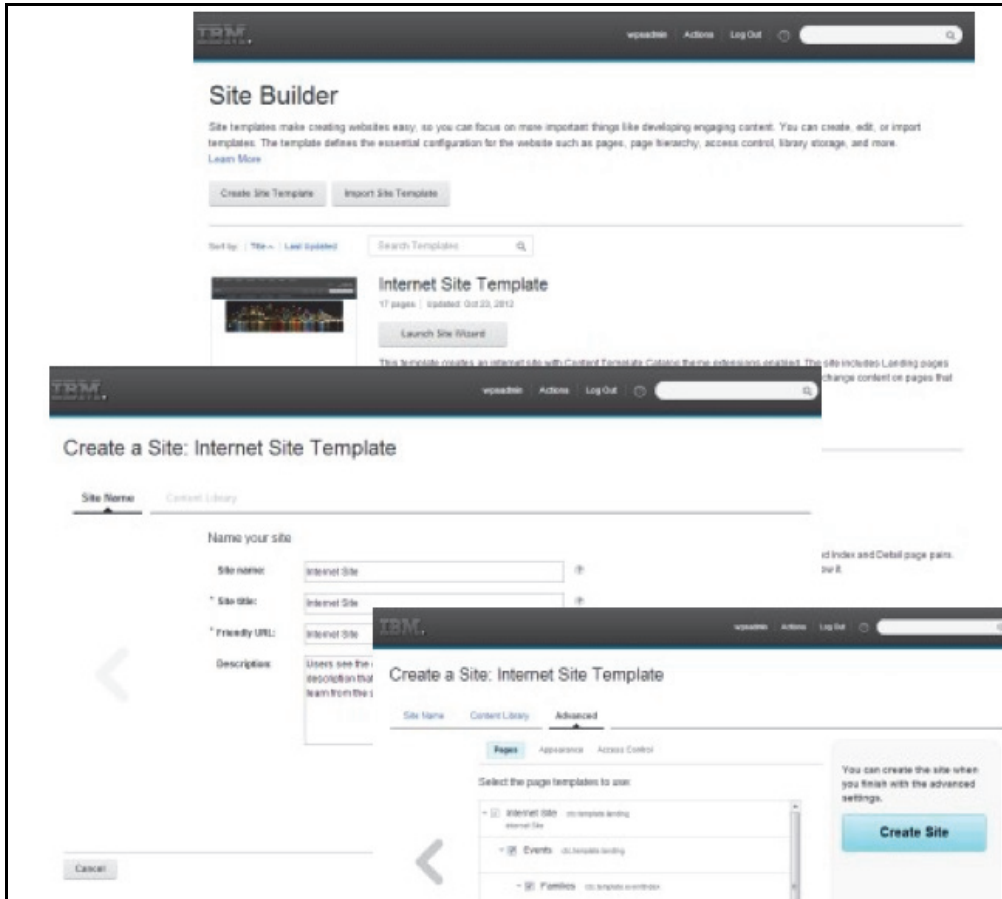


Figure 61. With Site Builder, IBM strives to offer some automation to users launching new sites in a more painless manner.

With respect to multisite management, IBM offers Site Builder, where an appropriately permissioned user can create departmental, team, extranet, or external-facing websites in a relatively quick, push-button fashion. The Site Builder can help marketers who need to rely on some automation in creation and support of, for example, landing pages and microsites — without the need to rely on developers, which you would otherwise fully expect with IBM WCM. The site templates, however, initially do need to be developed by your IT department. Developers can also restrict the depth of available customization points, such as modification of URLs, pages, content, and styling. A related feature to this is “Projects,” which you can use to build, modify, and deploy websites collaboratively with other teams in your organization.

Thus far, we’ve discussed cloning, but what about managing multiple sites on an ongoing basis? You’ll want to investigate carefully to see if WCM’s portal-based constructs can do

what you need. WCM enables content authoring and rendering to be scoped to Virtual Portals, which are logical instances of WebSphere Portal that run off one physical instance. Each Virtual Portal can be associated with its own LDAP realm, or they can share realms. You can think of a realm as an arbitrary combination of designated branches from an arbitrary number of LDAP directory trees. Content libraries can be scoped to Virtual Portals, and access controlled by realm. This can come in handy in use cases where a reseller extranet and a customer extranet need to share certain items, but have unique access-control requirements. Instead of setting up (and paying for) multiple instances of WebSphere portal (and creating sites for each — and trying to manage the dependencies among them), you can run virtual instances with dedicated, walled-off security realms — all off a shared resource base.

It sounds good in theory, but to get optimum performance from this kind of “virtual overlay” environment requires an investment in setup, tuning, profiling, and troubleshooting. Proceed with caution.

Overall, IBM Web Content Manager feels slower moving than it should be from an R&D perspective. Note that WCM is somewhat unusual among IBM products (but of course, remember that it was originally not an IBM product) in having relatively limited Java APIs, at least at the repository-services level — although IBM improves this with each release. Basic services for creating, saving, or deleting the dozen or so most common content components are exposed adequately. A relatively sparse JSP tag library is provided for component retrieval. REST APIs are available for certain user- and group-management tasks. The repository implements JCR (although it’s doubtful your developers will want to touch it directly). The newly added CMIS support may allow another option for HTTP access to repository services.

The WCM integration story (featuring WebSphere Portal at every turn) still feels like old-fashioned vendor lock, a situation somewhat improved by recent updates such as CMIS support, but IBM argues its customers prefer this stack strategy. Maybe existing “IBM shops” do, but we’re not so sure. In any case, evaluate the product carefully regarding your needs.

Performance

Historically, WCM has proven to be quite resource intensive, sometimes bogging down with just a handful of concurrent contributors, although at least one large customer reports being able to support 1,500 concurrent users. Large installations will almost certainly require an investment in multiple-processor machines, which could become a bit painful since the product is licensed by CPU (by Processor Value Units) under a complex licensing formula that varies by chipset.

If you’re planning to have hundreds of contributors and you need clustering with automatic failover capability, you should set aside an ample budget for hardware (which also means an ample budget for WCM licenses, which are sold by the CPU). Like OpenText, CoreMedia, and Adobe Experience Manager, this is not a light-footprint system.

One consultant characterized delivery-side caching as very basic, noting, “It can’t even ‘expire’ objects in its own cache, when a content author makes a content change.” Another developer told us that the caching is not extensible. Invest time in order to understand how caching works, especially with respect to factors like setting cache criteria and invalidation, which reside more in Portal than in WCM.

IBM has attempted to address some of the product's long-standing performance and stability shortcomings. Nevertheless, you should bear in mind that portlet-based pages are assembled in real time, every time they are visited (as required by the rules-based personalization of the portal), which means there is a certain amount of unavoidable latency — even with caching implemented.

Content

Contributor Experience

Overall usability sees incremental improvements almost every release, but there is still a great deal of clicking involved in order to complete common contributor tasks.

IBM's “Homepage” feature helps users get a better overview of tasks in progress, allowing a view on ongoing “projects” (content items in a form of creation workflow) and other portlets containing useful information updates. While this is hardly an innovation, it does help IBM pull its offering up toward the shoulders of its enterprise peers.

As with other IBM products, WCM has accessible interfaces, and IBM has localized the interface into 31 languages, the most among all products in this report.

Content	
Contributor Experience	
Overall Usability	●
UI Accessibility	●
Contributing Content	
Authoring & Transformation	●
Tagging & Taxonomy	●
Content Reuse	●
Media & Document Management	●
Repository Services	●
Content Lifecycle	
Workflow	●
Globalization	●
Archiving & Compliance	●

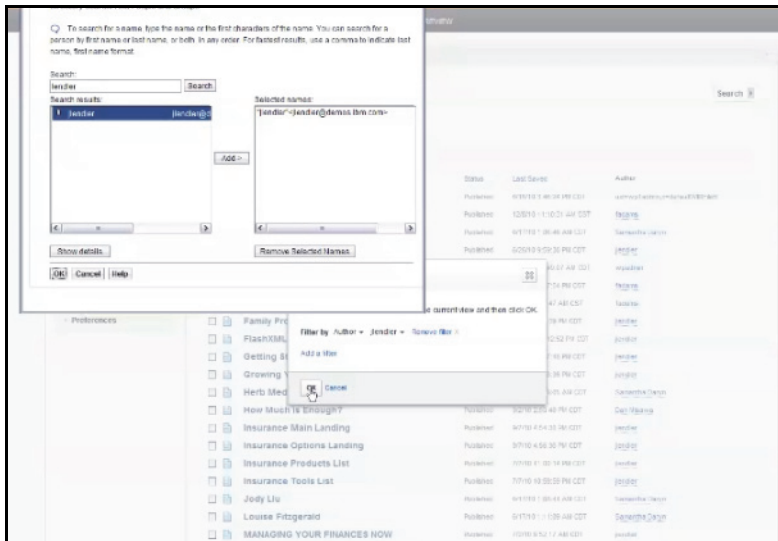


Figure 62. There are still quite a few pop-ups and clicks.

Contributing Content

Authoring capabilities come via a standalone portlet in WebSphere Portal Server (WPS). The JSP files that make up the interface are saved separately on the file system, and are not

managed within the system at all (unless you set the system up that way). Thus, you can customize the interface, but you'll want to oversee configuration management very carefully. Developers tell us that customization is not trivial.

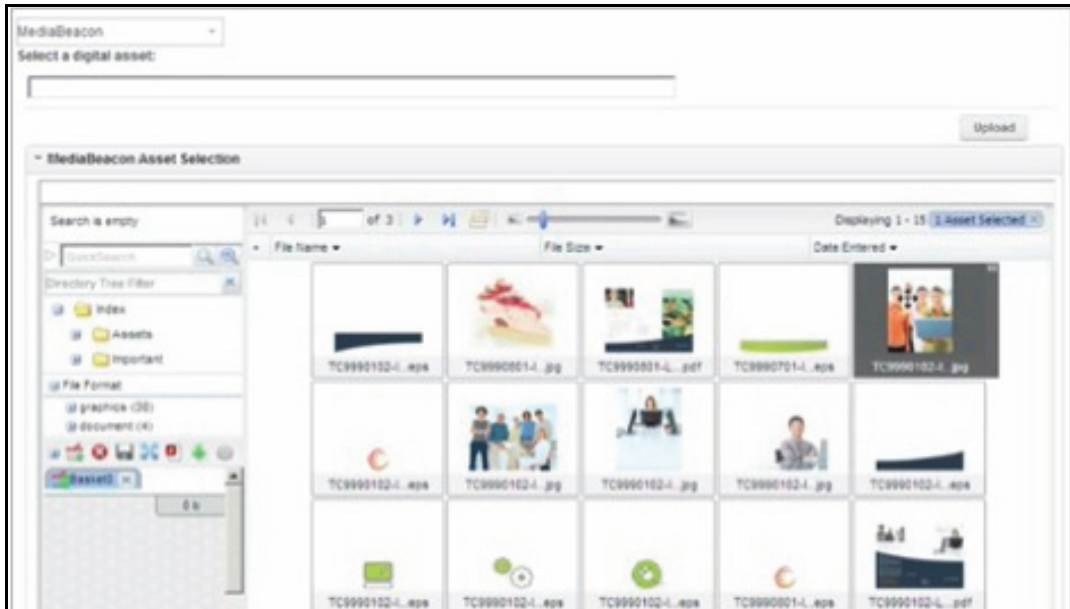


Figure 63. IBM stores and manages media assets in the built-in Image Library.

Rich media management in IBM WCM is often accomplished using the built-in Image Library. That functionality suffices for companies using their WCM as a “light” DAM application. For more sophisticated DAM and DAM-like needs, IBM started offering an integration with MediaBeacon’s R3volution Enterprise product (under the IBM Rich Media Edition moniker), which we extensively review in our [Digital and Media Asset Management](#) research. R3volution is heavy-lifting, enterprise-grade digital asset management software that organizes and manages digital assets (e.g., images, videos, and audio files). There’s a catch here, however; you integrate MediaBeacon through the Portal layer, rather than directly with IBM Web Content Manager, which could prove clunky in practice. IBM claims the integration also is available via the runtime or authoring templates in the content authoring environment; investigate this claim if this OEM’d functionality is important to you.

For repository search, the UI enables you to apply filters to narrow your results. You can also select from many predefined views. IBM says the UI helps reduce clicks and is more efficient. While there’s certainly an improvement compared to earlier versions, it still leaves much to be desired. For example, if you have to filter content by author, you first open a popup to select the author. Another popup opens and you search the author and add that as filtering criteria. That’s quite a few pop-ups — as well as clicks.

WCM “Projects” help you combine related changes into a single collection as part of a single update. However, most other major WCM tools already have these packages.

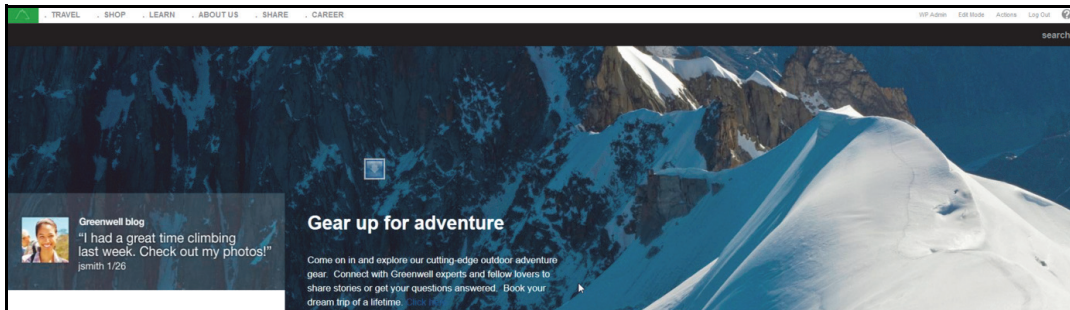


Figure 64. In-context editing in IBM Web Content Manager: editable functions are indicated by icons on the page, and with a menu in the top right corner.

With the WPS delivery environment bundled as part of the solution, WCM can enable in-context editing very smoothly. There are some other nice portal-like features. For example, authors can save frequently modified content items in a “my items” directory. The product comes with a compliance-checking tool, and it will manage internal links as objects, alerting authors when targets have moved.

For rich-text editing, IBM OEMs the Ephox EditLive! editor. On the plus side, the Ephox editor is quite feature-rich, with powerful table-creation UIs, the ability to check for W3C-compliant HTML, built-in accessibility checker, automatic spell checking as you type, an equations editor, some inline image-editing capability, version-compare, and a lot more.

On the down side, EditLive! is a Java applet (and a sizable one that takes a few seconds to load), which means — among other things — that content contributors need to be provisioned with a client-side JRE and Java browser plugin. This doesn't have to be a showstopper, but if your organization has never mass-deployed Java applets before, be aware of the potential for a bit of help-desk pain (at least, initially). Additionally, you'll have to decide how to deal with incessant JRE auto-updates, which can be hard to turn off. As an alternative, it is possible to swap it with another non-applet-based editor, but if you do that, you may compromise on features.

Additionally, while Ephox EditLive! can be configured to allow users to upload multimedia objects from a local file system, IBM's File Transfer applet doesn't allow the import. Thus, if you want users to have the ability to embed audio or video objects, you'll need to create a workaround to add them separately. Also, the Ephox “Broken Hyperlink” checker does not work with the IBM internal hyperlink structures. Again, a workaround would be required to validate hyperlinks. Alternatively, you can use the JS-based CKEditor.

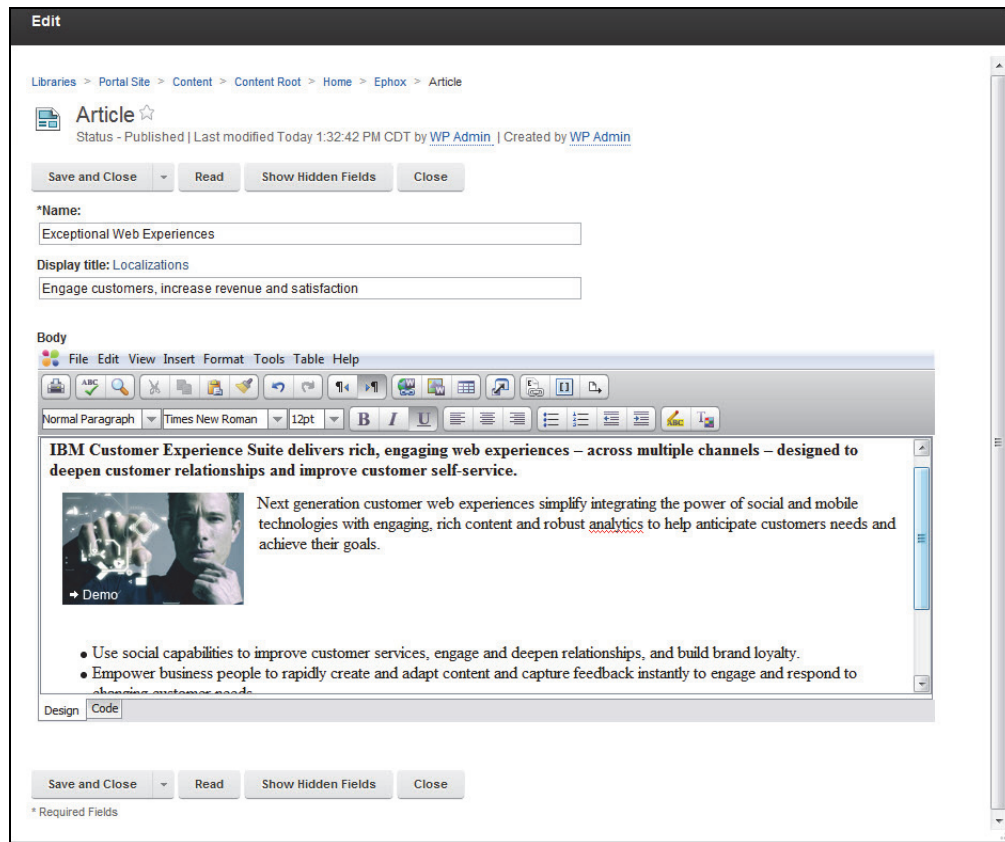


Figure 66. The authoring interface uses the Ephox EditLive! WYSIWYG editor, which is feature rich, but Java-applet based.

IBM WCM offers the ability to ingest content using WebDAV and Web Content Integrator. Using WebDAV, you can save your content to the server directly from Word. Remember though that WebDAV should be used only for occasional editing of web resources (properties files and web pages), and not really for documents. The Web Content Integrator (WCI), on the other hand, allows you to get content and metadata from other systems using RSS to WCM. You can then use Portal’s features to manage and display this content.

To create new pages from the main WCM interface, contributors first pick the content type, then add content, and select a site area. By picking a site area, they are also setting the default output layout, which is determined by site area. It is therefore possible to pair different page designs with one content template in different site areas. This is good for basic document repurposing, but more granular reuse of content elements (i.e., at the level of assembling a complex XML document from numerous fragments) requires substantially more development effort, and previewing content in multiple renditions is, by default, a manual process. In short, this is not a good platform for content reuse.

The “Project” function is another route for contributors to create new pages or sections of websites. A workflow task is created, which allows you to create or assemble content items against a pre-determined set of site templates from which you produce pages. This potentially could be used for devolving tasks from site managers to individual contributors, creating a workflow task for completion before any content has been produced; for example, “Create a

press release for this new product.” The subsequent content creation process would then be managed within the project workflow.

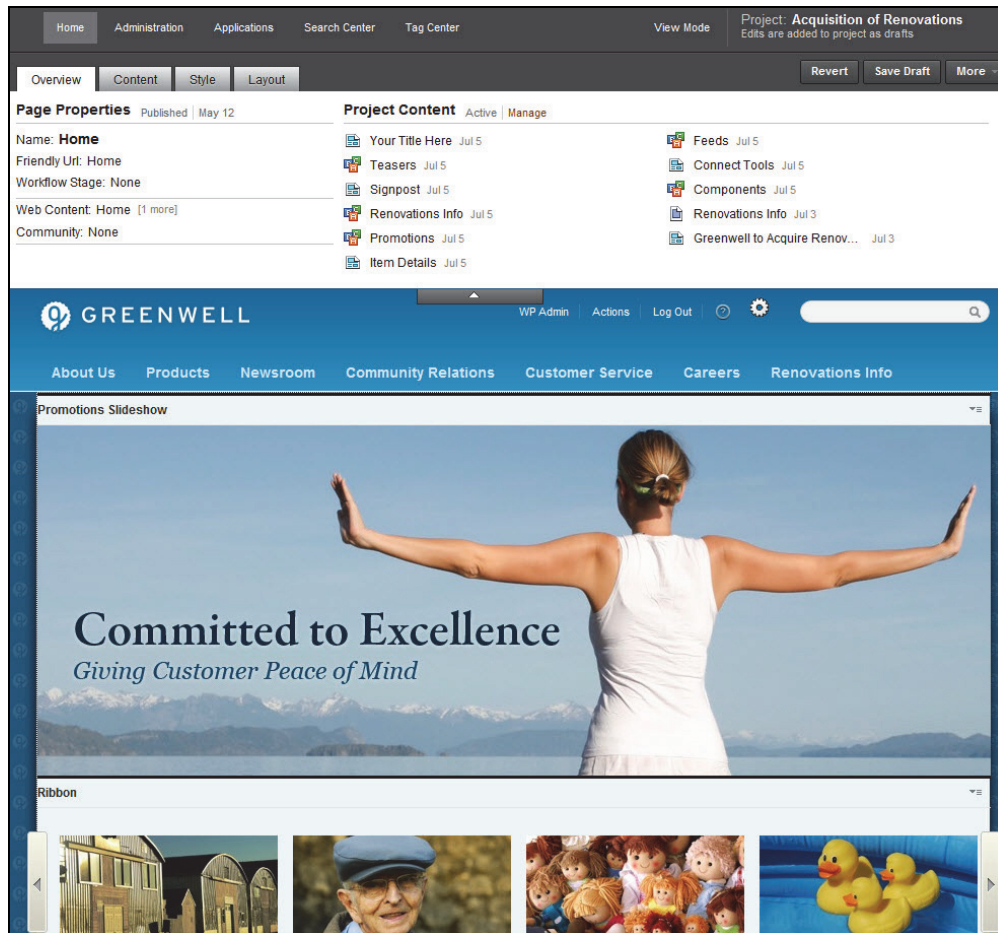


Figure 67. Using the “Project” functionality to build a new page.

You can also use Web Content Viewers — a set of portlets — for rendering WCM content, and IBM supports JSR-286.

One interesting and potentially very useful component is “browse by taxonomy.” To its credit, WCM has an unusually rich taxonomy model. Through a browser-based interface, you can move trees and add nodes to the taxonomy. For metadata-rich sites, where classifications are driving navigation, you can move entire site areas by modifying the taxonomy structure.

Another way to find what you are looking for is by way of tagging and ratings. Tagging allows you to establish a folksonomy with random text to tag an object. Rating is like voting; you (and other users) rate an object on a given scale and the combined rating is a set of stars. Note, however, that IBM is behind the times; star-based ratings have been replaced in most enterprise settings with “thumbs-up” or “plus-one” rankings (SharePoint has a similar problem). You can tag and rate a page, portlet, or even content within a portlet.

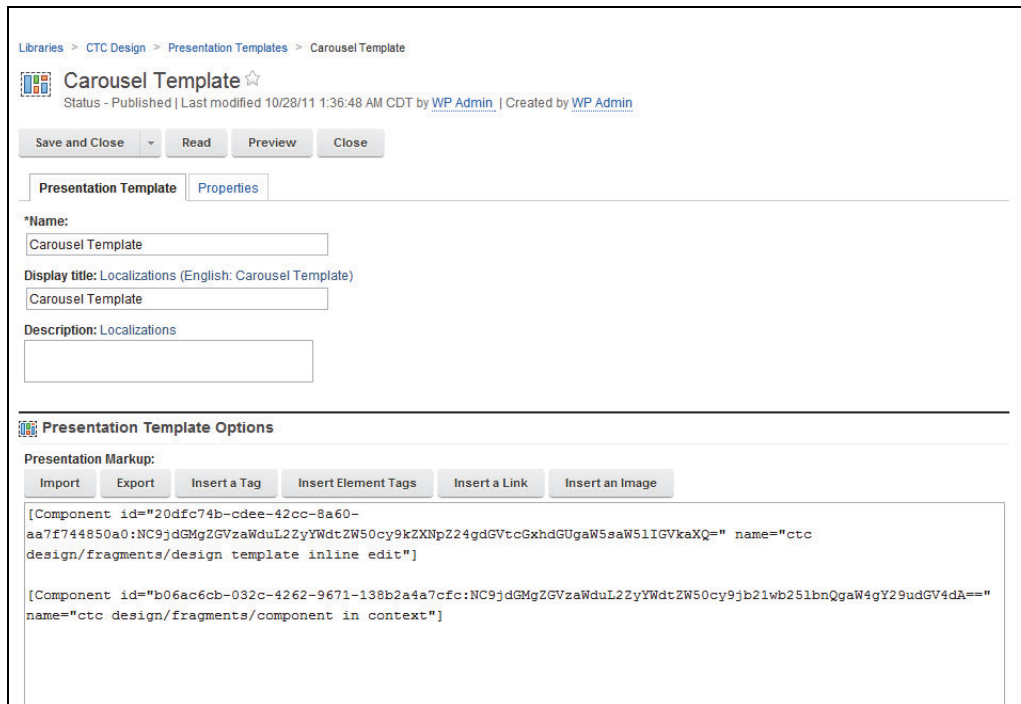


Figure 68. Adding elements to a page via the script library.

You can give tagging and rating privileges based on users but not content type. What this means is that you can't restrict users from tagging anything as long as they have permissions.

Content Lifecycle

Workflow in WCM is a mixed bag. On the plus side: Depending on workflow settings, authors can publish to different target environments (e.g., staging for QA), and there are nice facilities for virtualized preview in different stages. The product originally was designed for very simple approval workflows, and IBM has evidently not paid much attention to this part of the system since taking over the Aptrix code. However, in the new version, IBM finally added a proper editorial in-box and suitable task notifications for editors. In addition, the workflow API has been expanded and the system supports custom actions, opening the doorway to potentially sophisticated BPM-like flows. Things get very complex from here, though, and you won't find UI help.

The product incorporates a "multilingual framework," intended to ease the rollout of multi-language site editions. It comes with documentation and an installation script to get you started. It demos well, but you should exercise diligence here before making any decisions — particularly if globalization is an important part of your web content management strategy. IBM claims to support multilingual customers by using the Clay Tablet partnership to facilitate bulk translation, but in reality the Clay Tablet functionality offers only subset of features needed for holistic multilingual content management.

My Web Content Home Page ?

Create and manage content for the site using the links below. Use the library explorer to add favorite templates and locations so that they appear on your home page.

[New](#) [Open the Library Explorer](#)

Create Content

My Favorites:

[New](#) [Press Release](#)

My Recents:

[New](#) [Alert](#)

[New](#) [Social Test](#)

[New](#) [FAQ](#)

[New](#) [Social Network Configurat...](#)

[New](#) [News](#)

[New](#) [HTML Content](#)

[Select from all templates...](#)

Open Views in the Library Explorer

My Favorite Locations:

[Open](#) [Social Configurations](#)

[Open](#) [Greenwell Press Releases](#)

Web Content Activity

Show: [All](#) [My Favorites](#) [My Recent Items](#) [My Drafts](#) [My Pending Approvals](#) Display:

[Open selected view in the Library Explorer](#)

- Your Title Here**

Moved by user WP Admin - From stage Templated Draft Stage to Templated Publish Stage | Greenwell CTC / Home / Renovations Info | Jul 5, 2012 | Status is Publish Pending | [Open](#) | [Preview](#)
- Teasers**

Draft moved to pending publish state by user WP Admin | Greenwell CTC / Home / Renovations Info | Jul 5, 2012 | Status is Publish Pending | [Open](#) | [Preview](#)
- Signpost**

Moved by user WP Admin - From stage Templated Draft Stage to Templated Publish Stage | Greenwell CTC / Home / Renovations Info / Components | Jul 5, 2012 | Status is Publish Pending | [Open](#) | [Preview](#)
- Renovations Info**

Draft moved to pending publish state by user WP Admin | Greenwell CTC / Home | Jul 5, 2012 | Status is Publish Pending | [Open](#) | [Preview](#)

Figure 69. The default workflow inbox.

Experience

Publishing

The native portal URLs are ugly, but WPS has tools for mapping them to friendly URLs. This can be tricky to set up initially; test it carefully.

On the delivery side, you can choose to run your site fully dynamically out of the WCM engine (in concert with WPS), or you can publish to the Portal and take advantage of that system's personalization capabilities, caching system, and rules engine. On the other hand, some have found WebSphere Portal personalization and caching to be less than impressive.

For multichannel and mobile publishing, IBM offers the so-called “Responsive Design Theme,” which allows you to deliver content across a variety of predefined devices, such as tablets and smart phones (iOS, BlackBerry, and Android). You can tap into the “Responsive Content Catalog,” which allows you to repurpose content across multiple mobile devices from a centralized repository. The previously mentioned Web Experience Factory IDE also includes some responsive components.

Note that there are plenty of customers who chose best of breed tools to do mobile web and native app development (as opposed to the WCM itself). You should evaluate your use cases and requirements carefully before choosing the appropriate tool(s) for your needs. In fact, IBM itself may push to sell you its separate, heavy-duty “Worklight” mobility platform.

Digital Marketing

With respect to personalization, WCM along with WPS allows business users to match site content to visitors using either business rules or collaborative filtering. The WebSphere Personalization service includes a browser-based rules creator that an experienced digital marketing would use to create rules, campaigns, and mappings. A rules engine executes the rules created here. It also includes the LikeMinds recommendation engine to evaluate rules for collaborative filtering.

Based on these major components, you could have simple filtering (such as displaying content based on user group), rules-based filtering (such as show specific content to visitors who are in a specific age group from a specific geography), or collaborative filtering (such as show what others bought, based on what a user is browsing).

The main drawback of the Personalization interface is its usability. Every click leads to a page refresh, so if you are writing a sophisticated rule with many conditions, it becomes quite clumsy. The personalization interface itself is a portlet-based interface. It can work on WSRP portlets, but the personalization portlets themselves cannot be used as WSRP portlets.

Experience	
Publishing	
Standards Adherence	
Multichannel	
Mobile	
Digital Marketing	
Site & Campaign Analytics	
Testing & Optimization	
Segmentation & Personalization	
Social Media Integration	
Promotional Campaigns	
Community & UCG	
Workplace	
Collaboration & Networking	
Dashboard	
Ancillary	
Site Search	
Online Forms	
Module Ecosystem	

In recent versions, IBM introduced incremental improvements and new functionalities for segmentation and personalization, with increased focus on enriching the granularity of attributes available. Nevertheless, IBM WCM lags behind its competition in this regard.

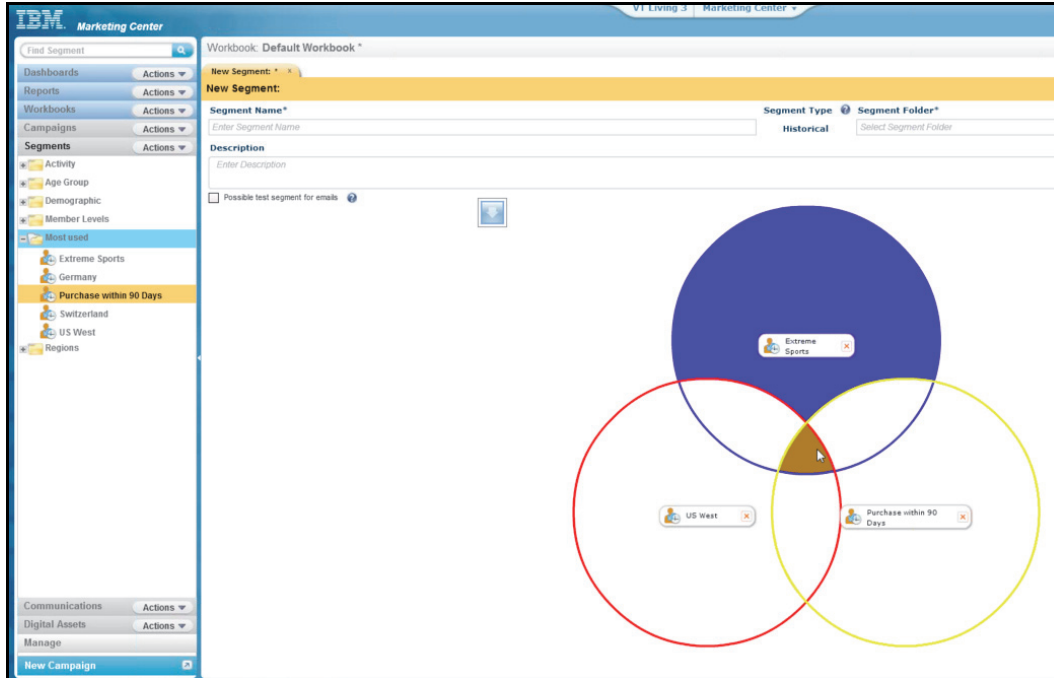


Figure 70. Assigning segments to a new campaign created in IBM WCM.

Social media integration within the publishing process is handled in a simple manner. For example, Twitter and Facebook are assumed to be publication targets. When publishing an update, it uses their public APIs to push customized versions of that content to the platform — then subject to a suitable workflow. Closing this loop to gain knowledge of the content’s success is restricted to basic “Like” or “Retweet” counts — unless you are publishing to IBM’s own “IBM Connections” product, where more detailed analytics become available.

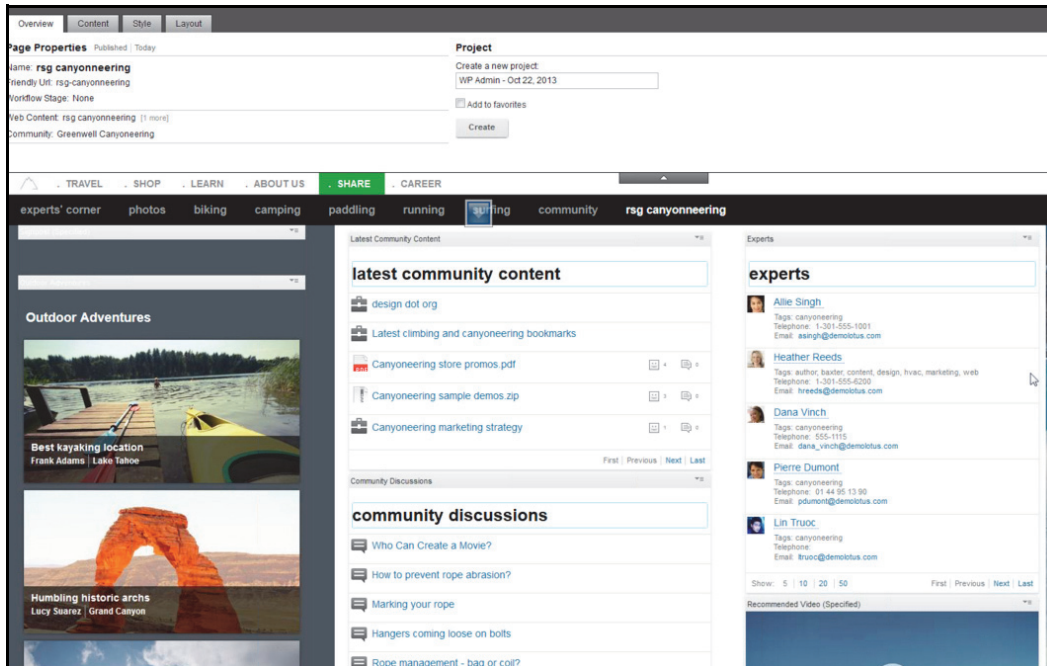


Figure 71. UGC, community content, and social sharing view in IBM Content Manager under Page Properties.

“Community Pages” allow subject-based content from Connections and other portal elements to be wrapped up together in a single page — albeit one that is manually constructed. It is a basic form of targeting, with the idea of it behaving almost as a landing page for visitors interested in a specific topic. In practice, it may involve a fair amount of manual setup, since each piece of content (page, portlet, etc.) must be manually associated with a specific community page. It’s not a complex process, but it needs to be associated with a content contributor’s daily role to gain the most advantage.

IBM’s analytics capabilities largely are based on the 2010 acquisition of Coremetrics, which the vendor is gradually integrating across the WebSphere estate — from its Commerce offering to WCM. In WCM pages, you can see in-context analytics overlays — much the way that other WCXM vendors have integrated (or OEM’d) their own web analytics packages. With IBM WCM, the current functionality is limited to page-/portal-level views with no notion of content-level analysis — except on the Social Analytics side, as illustrated in Figure 72.

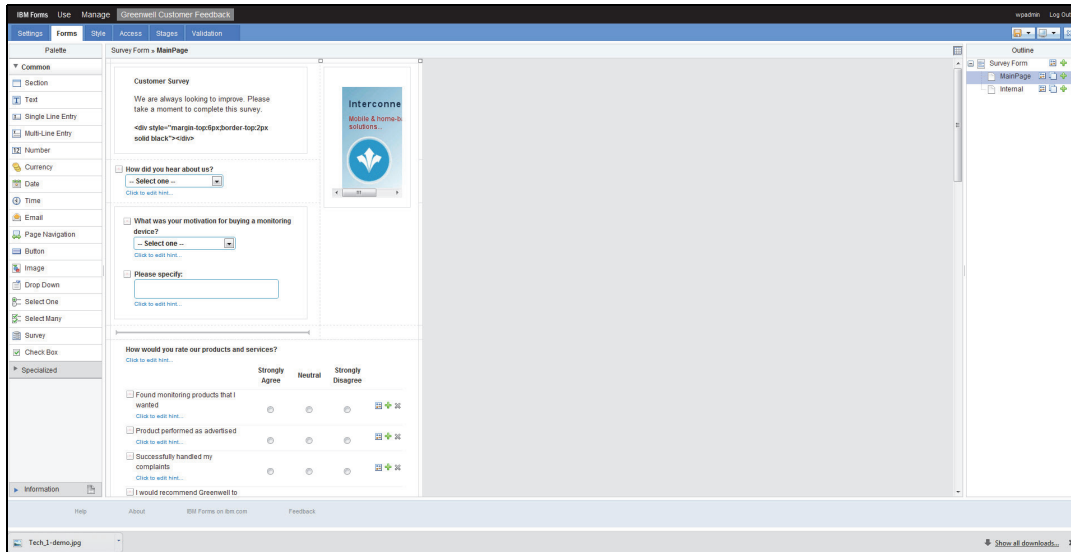


Figure 73. IBM's WCM forms builder.

IBM's online forms builder enables you to create and manage condition-based forms (based on users' behavior/history). The forms designer is envisaged to both produce traditional web forms as well as more advanced digital documents, the latter with the ability to be "digitally signed" by authenticated users (including those using tablet devices). As with all form designers, flexibility and capability comes with a high degree of complexity — especially when configuring conditional elements. If complex forms and documents are key to you — particularly if those elements are subject to frequent change — it is important to test the forms builder's creation and maintenance features thoroughly.

For search, the product relies on WebSphere Portal's Search Center functionality out of the box, but IBM offers bundle deals if you want its Apache Lucene-based Content Analytics engine (OmniFind Enterprise Edition) with WCM. Content Analytics is an elaborate, general-purpose enterprise search solution that's quite complex to administer. Most users will find Content Analytics to be overkill. On the other hand, Portal Search is not terribly strong by comparison, which is perhaps why IBM offers the Content Analytics bundle. According to one user, the native search facility doesn't recognize custom fields from an authoring template: "If I have a calendar template for creating events, and I have a start and end date and/or start and end time, I have no way of sorting the results, based on these fields." The built-in search also has scalability and consistency issues if you have more than one instance in a syndicated environment. This is an area to test carefully.

Vendor Intangibles

Product development is still largely driven out of the former Aptrix team in Australia. First-line support is global. Second-line support comes via call centers in the US and India. IBM's online support community for Web Content Manager used to be fairly vibrant. There is still some activity within these communities but they are considerably less active than what you might expect in the open source world, or even with other commercial WCXM vendors.

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

Consultants have complained to us of frustrations involving staying up-to-date on fix packs (not a terribly unusual complaint in the WebSphere world). Implementers say it can take too much effort to get what should be simple things to work as expected in the many-moving-parts runtime environment of WCM, and many seem to feel that more APIs is not a solution. Licensees pay a complexity cost for this or any large Java EE system, not just in initial implementation, but also with ongoing administration, maintenance, and support. Be sure to pencil this into your TCO calculations.

Conclusion

IBM's Web Content Manager product is priced at the high end of this tier, though you may be able to bundle it with a larger Portal license. Existing IBM clients should also see major cost savings through the company's "Passport Advantage" program, which provides discounts based on previous buying history. Big IBM shops can get WCM nearly for free — at least in terms of acquisition costs. At this tier, however, the initial cost of the software itself is usually the least of your worries; services will consume most of your budget.

WCM is a potentially powerful content management solution in the hands of experienced developers. However, from a sheer feature-richness perspective, WCM is still comparatively light for an offering in this price category. On the other hand, if you need an intranet-oriented solution that can scale massively (and play nicely with other IBM products), IBM WCM can perform that role quite handily.

Still, even the most loyal of IBM shops should not consider WCM an automatic buy — even if it comes "free" with your WebSphere Portal Server. At this price tier, you can get some interesting combinations of usability, extensibility, core functionality, and interoperability in competitive offerings. Shop around.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

OpenText: Web Experience Management

www.opentext.com

Vendor at a Glance

Specsheet	OpenText: Web Experience Management 8.1 Summary
Geography	Global
What's New	<ul style="list-style-type: none"> • New inline text, link and image editing feature introduced. • Integration with external content sources like OpenText Content Server, Media Manager, Social Communities, and MS SharePoint
Strengths	<ul style="list-style-type: none"> • Java APIs and J2EE architecture afford a broad platform support matrix, and a certain degree of developer friendliness • Optional "VBIS" development and runtime environment facilitates integration with a wide variety of remote systems • Wide array of additional modules available for purchase, covering collaboration, rich media services, community content, and more • Potentially very scalable, with a decoupled delivery model and wide array of deployment options for complex network environments • Licensing model is comparatively easy to understand, and bundle pricing is available on popular modules • Developers with OpenText WEM experience are not hard to find • Technology has found a solvent and profitable owner in OpenText
Weaknesses	<ul style="list-style-type: none"> • A potentially very complex product, requiring significant investments in development, support, testing, training, and administration • Somewhat page-oriented system with comparatively weak support for manipulating XML content, and product architecture is somewhat ill suited to component-level content reuse • Java-applet-based rich text editor presents maintenance challenge • Social software applications reside outside the core WCM product • For a platform of this size and maturity, the customer/developer community lacks coherence and visibility • Documentation not par with the product's capability and complexity • Customers report that upgrades can be painful given the company's history of not supporting previous APIs • OpenText is struggling to fit WEM into the rest of its software portfolio
Potential Fit	Global Enterprise, Ultra-Large Single Site, Advanced Marketing Portal
Unlikely Fit	All Basic Scenarios, Global Digital Workplace
Compare To	HP, Oracle, Adobe, IBM, CoreMedia
Operating Systems	Solaris, Windows, AIX, HPUX, Linux
Repository	Database: Oracle, MS SQL Server, DB2
Client	Browser-based (Firefox, IE and Safari), plus applet for rich-text editor
App Platform	Java: IBM WebSphere, Oracle (WebLogic, Sun), Tomcat, Red Hat JBoss

Specsheet	OpenText: Web Experience Management 8.1 Summary
Licensing	Per installation, starting at US\$95,000. Package pricing based on CPU and optional-module combinations — expect to start around \$250,000
Ownership	Public (NASDAQ:OTEX; TSX:OTC)

Summary

OpenText’s WEM (formerly Vignette Content Management) suite, which is well-known for its rich feature set and enterprise-grade architecture originally conceived at Vignette, has for many years symbolized a kind of gold standard in the industry. Regardless of how you judge the current technology, it’s hard to deny that Vignette lost momentum in the years preceding the current economic downturn. Vignette’s sale to OpenText came after a dip in maintenance revenues — the lifeblood of any established software company — which went from a steady decline to a potential free-fall.

VCM 8 was the first major release after OpenText acquired Vignette. In fact, OpenText released new versions of both VCM as well as the separate Portal offering. The update doubtless comforted customers who were concerned about the products’ future, but did little to resolve the fundamentally byzantine architecture underneath.

To be sure, V8 is built on the long-standing V7 edition, and that platform has significantly matured. Long-needed usability enhancements have finally found their way into WEM, and the range of integration options has never been greater. Prospective customers should understand, though, that this is a complex, evolving platform that requires significant investments in training, development, testing, and maintenance (not to mention hardware, software licenses, and support). Likewise, although various package deals are available at price points that are less than half of what you would have paid for all of the “separates” some time back, you should realize that a Vignette system — encompassing multiple optional modules — easily can sail past the half-million dollar (US) mark.

With release 8.1, OpenText has rebranded VCM as OpenText Web Experience Management (WEM) and removed the word Vignette on the presentation tier, website, and documentation — for the most part. Under the hood, however, the original Vignette code continues to exist.

OpenText’s WEM is best suited to Enterprise E-business scenarios that can take maximum advantage of its scalability, strong content integration features, and J2EE development-platform potential. Because of the system’s lack of comprehensive XML support and

Scenario Fits	
Simpler Site	
Informational	<input type="radio"/>
Microsites & Landing	<input type="radio"/>
Mid-Range	
Basic Digital Marketing	<input type="radio"/>
Mobile Site	<input checked="" type="radio"/>
Community Oriented	<input type="radio"/>
Complex Site	
Advanced Marketing Portal	<input checked="" type="radio"/>
Global Enterprise	<input checked="" type="radio"/>
Multichannel Publishing	<input checked="" type="radio"/>
Ultra-Large Single	<input checked="" type="radio"/>
Digital Workplace	
SMB/Departmental	<input type="radio"/>
Global/Enterprise	<input checked="" type="radio"/>

somewhat simplistic content model (which tends to inhibit granular content reuse), it is less well suited to scenarios in which highly compositional content models are important. As a development platform, WEM is surely overkill for simple or mid-range scenarios.

Customers should be particularly wary of WEM/Portal combination sales, since OpenText's roadmap calls for significant portal functionality to be migrated to the CMS tier — arguably something that should have been done long ago, but could prove to be painful for those who recently chose to invest in both platforms.

Introduction

Like most enterprise platform vendors during the previous decade, Vignette invested heavily in marketing and tended to sell to the higher rungs of the corporate ladder. The company succeeded in this tactic for many years, acquiring a long roster of marquee accounts, but increasingly faced a great deal of competitive pressure on price, features, architecture, APIs, social applications, and more from hungry new contenders who've muscled their way into deals that Vignette otherwise would have had sewn up.

The company also used its substantial cash reserves to acquire other companies, most notably Epicentric (Portals), Intraspect (Collaboration), and TOWER Technology (Document and Records Management). Early in 2008, Vignette acquired Vidavee, a company with expertise in streaming video technologies.

With the 2003 release of V7, Vignette maintained its traditional focus on providing both content management and delivery functionality, but it began doing so through two separate products: a WCM product (known as VCM), and a portal (Vignette Portal). The latest version has built upon this, and has brought in quite a few usability improvements. This report focuses principally on VCM now known as WEM — Web Experience Management.

In 2009, Vignette (after a series of layoffs and outsourcing exercises) was acquired by Canadian technology giant OpenText, an ECM vendor whose catalog already included the RedDot CMS product. The Vignette acquisition raised the immediate question of whether one CMS technology would be dropped in favor of the other, or whether the two technologies might be melded together somehow. OpenText continued its acquisition spree and subsequently acquired Nstein, a vendor with DAM and WCM capabilities, but known more for text analytics.

In the end, OpenText continued to invest in Vignette and has released newer versions of both WCM and Portal.

The core OpenText Web Experience Management platform capabilities are extended with the following bundled modules:

- Presentation Management Services — A combination of two former Vignette modules — Dynamic Site Module (DSM) and Dynamic Portal Module (DPM) — This is a set of pre-packaged, JSP- and portlet-based web applications deploy in the delivery tier. Formerly optional, it's now part of the core offering
- Media Management Services — Formerly Vignette Rich Media Services — Enables ingestion, transcoding, and delivery services around video

Optional modules for WEM include:

- OpenText High Performance Web Delivery — Formerly Vignette High Performance Delivery — Provides improved site performance through caching of static and dynamic components
- OpenText Web Experience Management Audit — Formerly VCM Audit — Enables auditing and reporting within WEM for all content management operations (including read operations, workflows, search) for compliance purposes
- OpenText Web and Social analytics — Provides web analytics for WEM-powered websites
- OpenText Social Communities — Formerly Vignette Social Media or Vignette Community Applications/Vignette Community Services — Provides community tools (tags, ratings, comments) and community apps (blogs, forums, wikis, image and video libraries, calendars, and idea management) for websites.

In addition, you can integrate WEM with the following OpenText products:

- OpenText Web Recommendations — Formerly Vignette Recommendations — An OEM of the Baynote technology for recommendations and social search capabilities, it captures user feedback in various ways, infers “intent,” and adds recommendations to aid with information access
- OpenText Campaign Management — Formerly Vignette Dialog — Enables interaction management by automating personalized sequences of customer dialogs across websites, portals, email, and other channels of delivery along with tracking and reporting
- OpenText Web Integration — Formerly VBIS — A lightweight development environment for integrating information sources and services
- OpenText Portal — An industry-standard portal platform that we cover separately in our Enterprise Portals Research. OpenText salespeople try to sell it hard with WCM deals, and it includes functionality that you might otherwise see in more highly coupled solutions
- OpenText Video Services — A SaaS platform that can be used to submit, transcode, and stream video content
- OpenText Content Server — Formerly Livelink — Can be used as a source of documents for WEM-managed content
- OpenText Media Management — Formerly Artesia — Enables sourcing of media assets for WEM content
- OpenText Content Analytics — Formerly Nstein — Used for automatic or manual metadata extraction for WEM content (based on the semantics)

Note that although WEM 8.1 introduces integration with OpenText Content Server, Media Manager, Social Communities, and Microsoft SharePoint 2010, at this point, you can only import content from these sources into your WEM repository.

If you already have Vignette 7x, you can upgrade to V8.1 and still keep the V7 UI. This is a nice feature and it is useful if you want to roll out a new system to many users gradually. You can also use V8 in compatibility mode, which means that the old sites remain in V7 mode and the new sites are in V8 mode. However, there is no direct migration path from V7 to 8.1; you will first need to upgrade to 8.0 and then to 8.1. For the most part, this migration should be smooth, however if you have used V7 API to develop custom widgets, be aware that this can be a complex and time-consuming process. Customers indicate that the V8 APIs are not backward compatible, and custom widgets written in V7 no longer work in V8 resulting in code rewrite. Before you plan for this migration, we advise that you gauge the extent of API usage in your implementation to understand the risk, check for the latest updates and patches, and plan for the migrations appropriately.

Technology

Technical Administration and Security

WEM is comprised of logically separate stages. Think of a stage as a compartment for your content management activities (creation, approval, and testing). Usually, there is one Management stage and multiple Delivery stages (like testing, QA, and production), and you publish assets from one stage to another during the content’s lifecycle.

A stage can have one or more of the following:

- **Content Management Services:** Consists of modules for content creation, configuration, administration, and so forth. All of the content management functionality is provided by WEM server, which is part of its Runtime services. These Runtime services include an embedded Java application server, lightweight LDAP directory services, and JDBC drivers. The Content Management Services also provide a content API for manipulating content and are associated only with the Management stage.
- **Process Workflow Modeler:** This is essentially a Visio add-on to graphically create and manage workflows. It is a separate component that runs only on windows environments. Workflow modeler can connect only to the Management stage.
- **Content Delivery Services:** Consists of facilities for deploying and publishing content across different stages. This is done using deployment agents that read the deployment rules and move content to and from file systems and databases. Content Delivery Services are part of all stages — Management, as well as Delivery

Not all stages contain all services. For example, the Content Management Services are available in the Management stage only when the Content Delivery and Presentation Management services are available across all stages to “preview” the sites, as per the content in the various states (testing, QA, and production).

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

Like many other products in this report, WEM stores content from each stage in database tables (sometimes a single content type is split across tables) and file systems. WEM supports Oracle, DB2, and Microsoft SQL Server databases.

By default, WEM uses its own built-in LDAP for security and user management, but you can use your own LDAP server, as well. WEM supports IBM Tivoli, Microsoft AD, Novell eDirectory, OpenLDAP, and Sun's directory server. If your LDAP supports only SSL, then there are some special considerations; test carefully.

For databases, WEM supports IBM AIX, Solaris, Windows, and many flavors of Linux. It also supports VMWare if you prefer to use it in a virtualized environment. The good thing is that you can have a different operating system for content management and delivery environments; the not so good thing is that the database has to be same across these environments, which may have some licensing implications.

For delivery services, WEM requires an application server and supports IBM WebSphere, Oracle WebLogic, Apache Tomcat, and JBoss.

On a positive note, WEM supports a looser coupling of production and delivery systems. The system not only permits, but *encourages* physical and logical separation of production, staging, testing, and outward-facing delivery tiers (or stages). The publishing process can be triggered manually, automatically (e.g., as part of a workflow), or scheduled. You could also have a custom publishing process that publishes to multiple stages.

A Deployment Agent (DA) is the process that is responsible for publishing and deployment. A DA is a part of Content Delivery Services (CDS) and manages the movement of files to the file systems and content to databases.

Today, OpenText provides different mechanisms for content deployment and delivery. There are at least four ways to handle delivery:

- **Decoupled delivery:** These facilities allow you to publish and push static content or individual objects to a separate delivery environment. Customers still find this a bit temperamental; test carefully
- **Build your own web application:** Using the Java API, you can develop your own application to access and deliver content (WEM now includes its SDK)

Presentation Management Services:

- **Dynamic Site Module:** This module consists of a collection of pre-canned JSPs and servlets that you can customize to develop your own web application that lives on top of the WEM repository. It tends to emphasize JSPs, beans, and tag libraries over standalone servlets, and some licensees have raised performance concerns. It now has a caching sub-system that takes care of some of these concerns
- **Dynamic Portal Module:** This module exposes the Presentation Management services (templates, pages, themes, layouts, and in-context environment) to other Portal products — most notably Vignette's own (others include WebSphere Portal, Liferay, Oracle Aqualogic (formerly plumbtree) and SAP NetWeaver) — whereby the portal can display content from the WEM repository in real time

If your RFP/Tender asks for more than just run-of-the-mill sorts of dynamic page assembly via JSP, it is likely that OpenText will bid its Portal product in addition to WEM. OpenText

salespeople seem to promote the Portal when prospective customers have specific requirements for personalization, granular component delivery, or multisite management. In addition to being comparatively lower cost, the Portal also demos very well, but in some instances, it is merely compensating for functionality that WEM lacks.

Note that the same content can have multiple “end points,” such that you could publish the same material in both static and dynamic formats, at different locales. In general, customers with mixed models of super-dynamic and semi-static information will find good flexibility.

The platform offers substantial flexibility around the mechanics of content deployment. You can deploy content to heterogeneous delivery environments using a variety of approaches, including FTP, email, and OpenText’s own SOAP-based transactional system. In response to customer requests, WEM offers a relatively deep “job control” panel to enable sysadmins to set up and track the health of various publishing and deployment jobs (Figure 74).

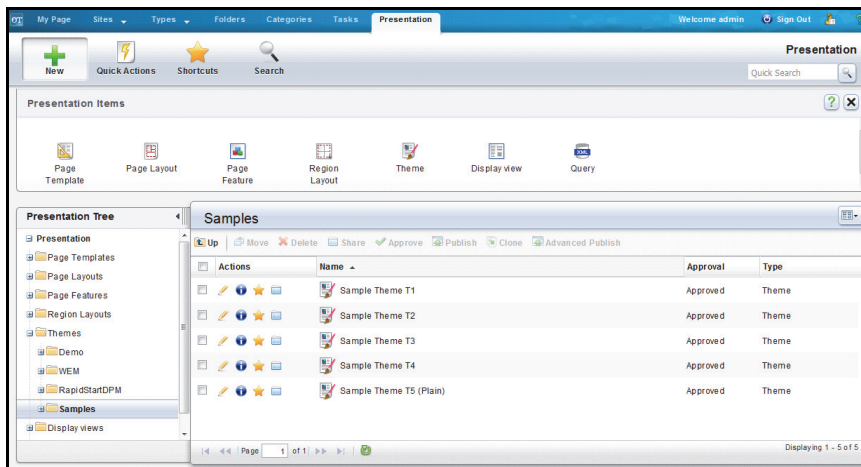


Figure 74. An administrator can view the status of all published assets from the management console and track progress.

Note that, like all transactional deployment systems, WEM requires a “receiver” module on the target delivery server(s). Some customers just purchase one receiver agent and then replicate content across the delivery tier using other means. WEM recommends multiple agents to avoid a single point of failure. Deployments can be triggered directly from workflow events. This is very useful in its own right, but it’s also important here, because final page assembly doesn’t happen in WEM until you publish (you can preview using your own XSLT and/or JSP templates in the meantime, or use either the dynamic portal module or dynamic site module for previewing).

Although WEM maintains its own access control lists, authentication occurs via an embedded lightweight LDAP directory. You can also use your own external LDAP. Like most of its direct competitors, the product performs real-time authentication at the beginning of each session (i.e., it does not cache credentials locally — but only during the session), which is a good thing.

For reporting (and auditing), OpenText offers WEM Audit (Figure 75), which provides the ability to track and document changes to content. WEM Audit comes in two flavors — a complimentary version that is part of WEM console and is available to all WEM customers. A

more advanced version, called “WEM Audit Enhanced” is a separately licensed module. The complimentary version includes basic event history views while the “Enhanced” version adds sophisticated features like dashboards, drill-down reports, and the ability to export report results.

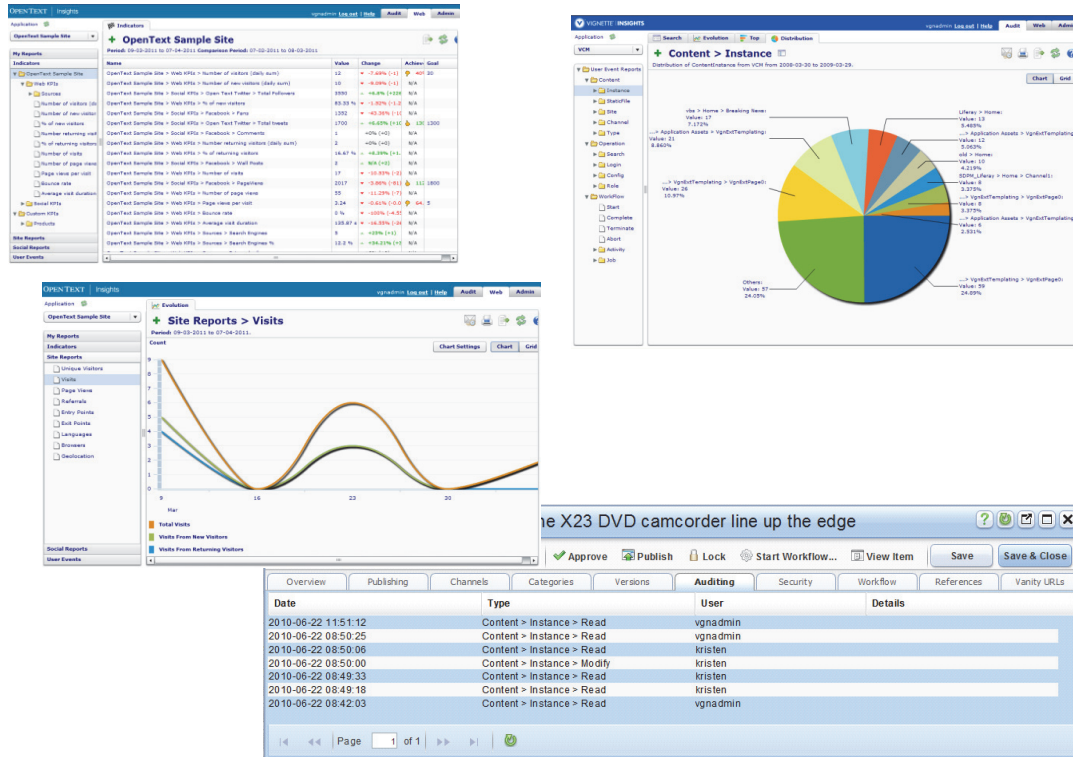


Figure 75. WEM Audit Enhanced console.

Development

Initially, there was no templating or renditioning subsystem in v7, and no way to pre-generate static content. This has improved and now you can assign content types to multiple output formats, and in some cases, preview that content in-context, and pre-generate static HTML using filenames of your choice. You can now create templates as part of a custom Java application, or as part of one of the Presentation Management Services (formerly Dynamic Site Module and Dynamic Portal Module).

A WEM Template consists of “fixed regions” and “managed regions.” Your developers create fixed regions that cannot be changed. However, your business users can manipulate (add, remove, or reposition) managed regions. Depending on your choice of presentation tier, you create these templates (along with associated layouts and themes) using taglibs and a combination of standard technologies like JSP, portlets, XSL, CSS, JavaScript and AJAX.

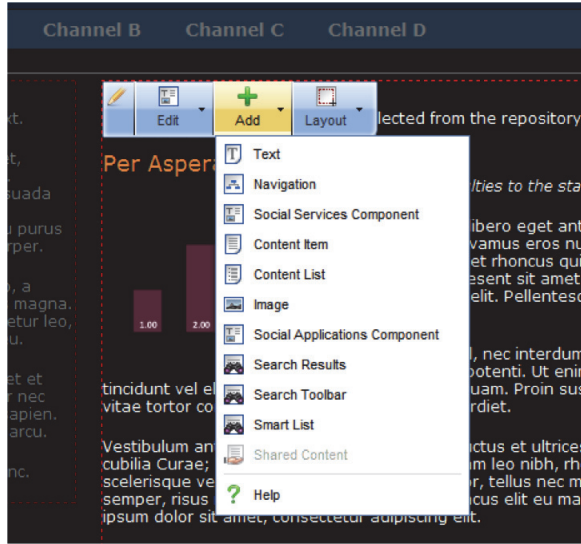


Figure 76. Assigning elements to a page.

In the templating system, business users can assign components to pages (Figure 76). OpenText offers a few pre-built components out of the box, but not as many as some of its competitors. The actual creation of layout of components on the page falls to a JSP developer using its tag library. However, business users can choose the layout they want and select the right components. This approach is similar to that of Percussion and other vendors, where individual elements must be crafted by designers, but final page assembly can become a business user function. Your preference here will ultimately depend on your operational model.

One traditional gripe among Vignette developers prior to v7 was that the server APIs were not robust enough, making it difficult to get at certain core features, especially in connection with library services. Vignette has worked to close this gap in V7, but with release 8.x, the gripe is back — this time due to the lack of backward compatibility of the APIs. At least one customer tells us that a goodly number of V7 APIs have been changed in V8; they have been either removed or attributes changed.

A key differentiator for WEM is its optional enterprise content integration (ECI) toolset called OpenText Web Integration (formerly VBIS, and before that, Oberon Prospero), focused on providing connectivity to legacy ERP and CRM applications, as well as unstructured content repositories. This allows you to draw content from them when needed and include the remote data in your web publishing efforts.

OpenText Web Integration is a Windows-based IDE that generates integration code in a variety of potential formats, including a mix of JSPs, and servlets on the Java side, or DLLs for Windows, or native Web Services. Some customers use Campaign Management (another module) to create highly customized email newsletters, which they then manage through WEM. If you anticipate that the majority of your web content will emerge from SAP or a Lotus Notes system, you should at least consider WEM — which boasts more than 70 different “adaptors.”

WEM 8.1 also brings some integration with OpenText Content Server, Media Manager, Social Communities, and Microsoft SharePoint 2010. You can search or browse content in these external sources, then import and copy the desired content into your WEM repository. This content can be imported either as files or instances of custom content types. (Figure 77).

Import Content		
External Content Sources		
Name	Type	Description
Digital Assets Content	OpenText Media Manager	OpenText Media Manager (Artesia)
Social Media Content	Vignette Community Applications ; OpenText Social Communities (VCA/VCS)	
Collaborative Content	OpenText Content Server	OpenText Content Server (Liveliink)
SharePoint Content	OpenText Content Server	Microsoft SharePoint 2010 (MOSS)

Figure 77. Importing content types.

One area where experienced content engineers may find WEM lagging is in XML support for content. With Web Integrations you can import and output XML content; but in terms of managing XML content in a way that facilitates composite-document creation and granular reuse of XML fragments, Vignette sticks to a straight relational model in which an XML document is treated like any other content item. If you intend to work mostly with XML content and have rather intensive reuse requirements, you should study WEM’s content model very carefully.

Developers create new content types that map to existing database schemas. WEM offers a powerful (and initially somewhat intimidating — though ultimately useful) UI to help map XML content structures (with inheritance relationships) to the database schema (Figure 78).

Name	XML Name	Bean Property Name	Type	Length	Widget	Searchable	Default Label	Summary	Visible	Required
ID	id	id	String	40	GUID	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Title	title	title	String	1000	Text Field	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Summary	summary	summary	CLOB	N/A	Text Area CCE	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Body	body	body	CLOB	N/A	EditLive! for Java	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Thumbnail	thumbnail	thumbnail	String	255	Content Select CCE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Image	image	image	String	255	Content Select CCE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Link ID	linkId	linkId	String	40	GUID	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Related Items	storyId	storyId	String	40	Content Relator CCE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Linked Item	relatedId	relatedId	String	40	Content Select CCE	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Sequence Num	sequenceNum	sequenceNum	Integer	N/A	Hidden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Group Key	groupKey	groupKey	String	255	Static Select CCE	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figure 78. WEM offers a powerful (if somewhat intimidating) UI for mapping XML attributes to database columns, giving the administrator or power user full control over visibility, searchability, and other properties for each mapped attribute.

Performance

WEM is a complex system and it can be tricky rendering reasonable performance. It provides guidance for installing it in different topologies based on your requirements. On the management side, you can have a multi-node cluster and can scale horizontally. In the delivery environment, a lot of performance related optimizations will depend on the application server you plan to use, as well as which delivery environment(s) you implement. Presentation Management Services have a cache implementation that is helpful in improving performance. By default, it stores cached content in memory and also stores a serialized version of content in the file system. In fact, there are quite a few caching implementations and options depending on which modules are in place; test it out carefully. In spite of what most vendors claim, caching is very difficult to implement properly, so pay special attention to how cache is invalidated when content changes and if you can cache fragments of content instead of a whole page.

For best results, you'll want to look into OpenText High Performance Web Delivery, or "HPD" (yet another add-on, representing caching technology that OpenText inherited from Vignette). HPD is actually fairly impressive in the way it staticizes pages with dynamic components. Like any caching subsystem, it's rather complicated to operate in practice, so you'll want to invest ample time and resources learning its nuances.

Content

Contributor Experience

Users have a choice of two interfaces for their content management activities. One option is to use the "Content Workspaces" interface (Figure 79), which is similar to the traditional management console.

The workspace consists of the following sections or tabs:

- **My Page** creates new content, your shortcuts, and saved searches
- **Sites Workspace** helps you organize content based on sites
- **Types Workspace** allows you access to content based on content type
- **Folders Workspace** allows you to access content based on a folder hierarchy
- **Categories Workspace** categorizes content based on the content classification you have defined for your site
- **Tasks Workspace** works with tasks and workflows
- **Presentation Workspace** creates and manages presentation templates, layouts and themes

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

Your users can either browse content by various dimensions like site, folder, category, and type — or use search capabilities to find relevant content. WEM 8.1 further enhances this capability by allowing users to file assets based on the content metadata (e.g., author or creation date) of the content type and timeline (e.g., modification dates). While this works well for contributors with small to medium websites, it can become problematic for high-volume enterprise users since the content displayed is not personalized. Specifically, if you have several thousand or millions of assets across several websites or website regions managed by several administrators, WEM does not restrict the asset display to only the assets that you’re authorized to view. So you could potentially end up viewing a large number of assets in your workspace that you don’t have authorization to access (not to mention edit). In addition, the V8.1 workspace and preview portlet do not support Internet Explorer 7 — although it was released long ago, it’s still extensively used on corporate intranets.

On the plus side, you can define your own shortcuts to frequently accessed sites, channels, or content items. There is a “Quick Actions” menu that enables you to create a new content item with default metadata already defined. The Quick Actions module is basically a place to create and store custom macros that store the content type, channel, and miscellaneous default attributes for the various content a user regularly creates. With WEM 8.1, contributors can share custom-created Quick Actions, shortcuts, and saved searches with other users. From within the workspace, you can also define new content types, start workflows, and do many other administrative and management tasks.

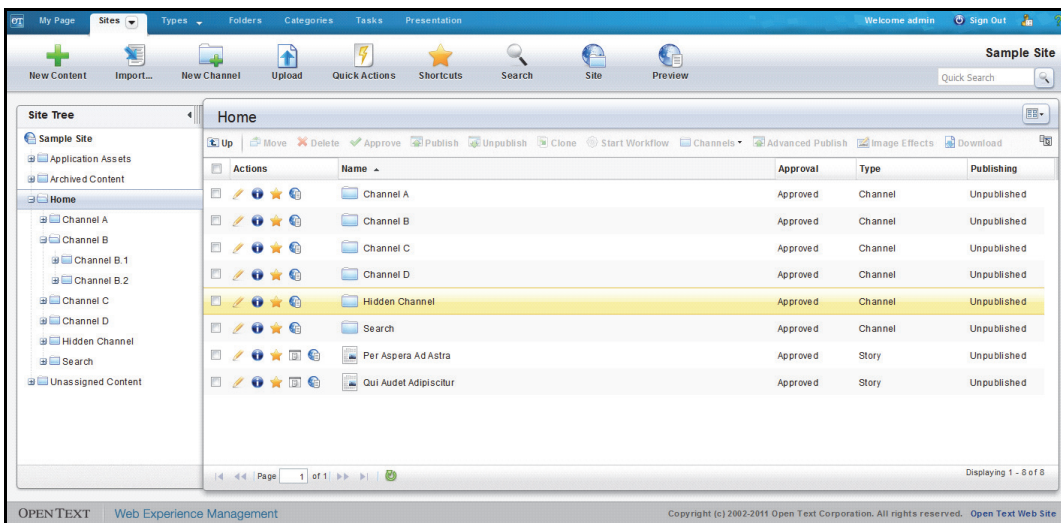


Figure 79. OpenText WEM 8’s content workspace allows you to manage content across multiple dimensions using a variety of views and hierarchies. A context-sensitive ribbon shows you the options based on the view you have selected.

The Search module allows you to search for content based on any combination of keyword(s), date, status, and content type. Rather than a page full of text links, your hits come back in a rich interface (Figure 80) which offers insight into each content item and allows you to take immediate action (Publish, Unpublish, Clone, Approve, etc.) on one or more selected items. Not immediately obvious is that you can also drag and drop individual items from the Results

list to the neighboring Shortcuts module of the Personal Home Page (so that you don't have to search for them again).

Note that repository search is now based on native OpenText search technology. The good news is that it's well integrated; the bad news is that OpenText is no longer investing major R&D in search technologies.

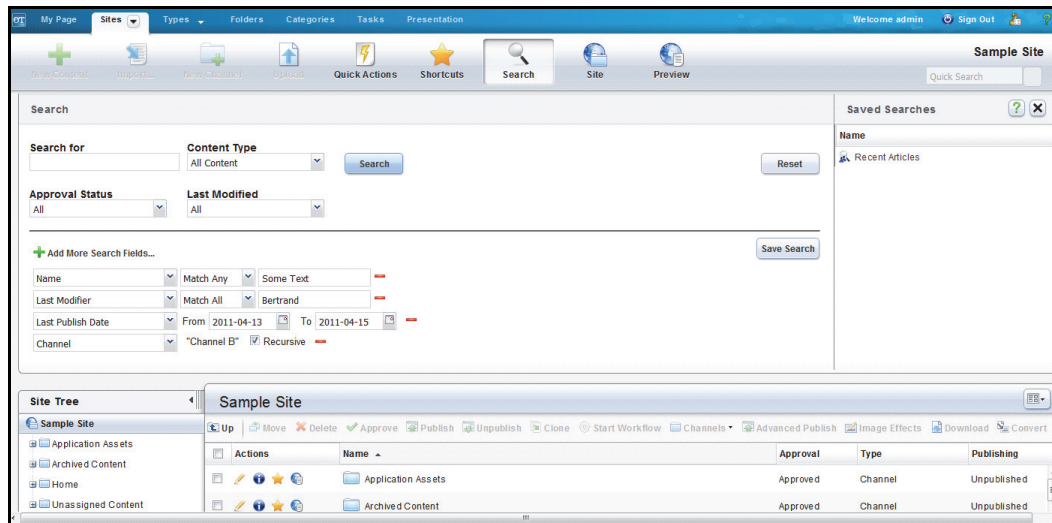


Figure 80. Search results are displayed in a rich UI that enables you to see information about each item, perform toolbar actions of various kinds on each item, and drag a given item from the search results to the shortcuts module so that you can reach the item directly without searching in the future.

When you select a specific item in any of the workspaces, it shows all the relevant information for that item using a properties palette. The palette has tabs for different sets of related metadata as shown in the Figure 81.

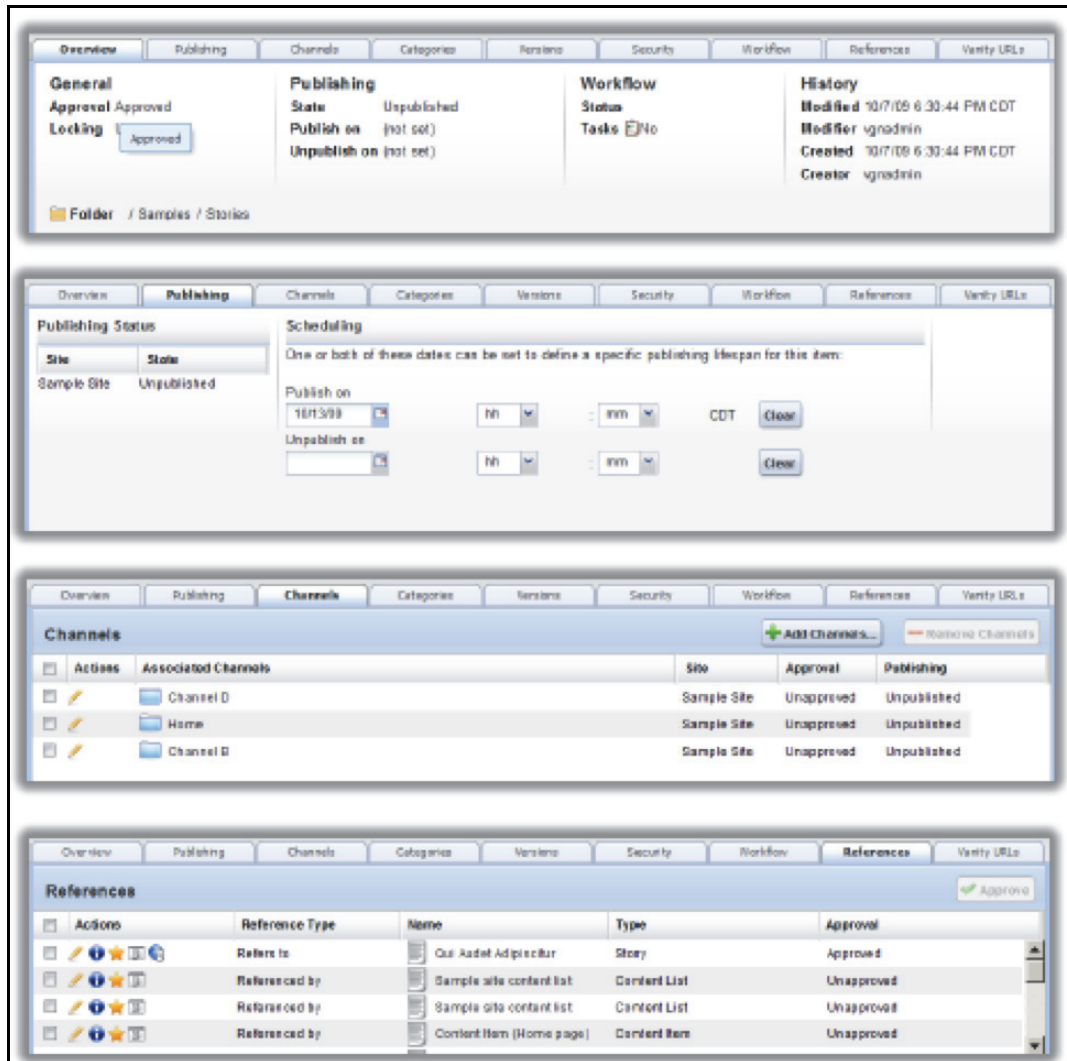


Figure 81. Selecting a content item shows its properties that are categorized and shown as tabs. The figure shows views for four such tabs — Overview, Publishing, Channels, and References.

Overall, the Workspaces console offers a very rich dashboard, but definitely one oriented towards power users who have been thoroughly trained to exploit potential efficiencies.

The other interface (new in WEM 8) is the preview environment (Figure 82). Using the preview environment, you can do many of the previously mentioned tasks in the context of the actual site. When you use the preview environment, a tools palette — a sort of a sidekick — appears on the page you are editing. This tools palette has features for you to change the look and feel, apply a different theme, create new content, and participate in a workflow or publish content. You can also add or remove components on the page using a component palette.

With 8.1, OpenText enhanced this interface to introduce inline-editing features to allow contributors to edit text and text attributes on the web page while previewing it directly

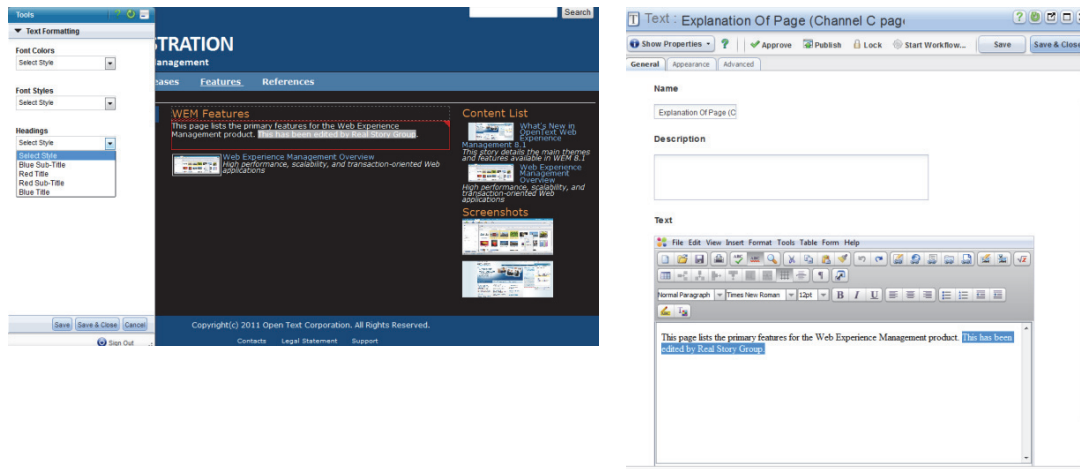


Figure 82. The preview environment provides for an alternative way to manage different aspects of content management. Inline editing features are only available in the Presentation Management Service.

The inline editing feature is only available if you are using the Presentation Management Service (one of Dynamic Site or Dynamic Portal modules), which helpfully is now bundled with the core offering.

This new way of managing content is very convenient for business users, but it has some limitations as well. Your users will need to spend a lot of time trying to figure out appropriate positioning for these palettes. Finally, creating pages in the context of presentation could have implications if you want to repurpose these pages for different destinations — something that WEM prides itself on being able to do. Note however that some of the productivity enabling features like shortcuts and quick actions are not available in this interface — yet.

WEM’s contributor interfaces have traditionally relied heavily on popup windows, tabs, modal dialogs, endless mouse clicking, and generally far too much “drilling around” to get any work done. There was also a lot of techie-sounding language in the default interface — the kind of language that happens when developers hand-code text strings in a UI. A lot of this has been improved in v8, and OpenText claims there are no more popup windows.

Contributing Content

For content authoring and editing, the product (see Figure 83) offers a forms-based interface, complete with Ephox’s applet-based EditLive! rich-text editor, which offers a great deal of power (as WYSIWYG editors go), but has the drawback of being a Java applet, thus requiring a client-side JRE (with all the attendant version-updates and help-desk complications).

You first select a content type for which you want to create a content instance. A forms-based editor then opens up with all the required fields (and the EditLive editor, when required). Note that even when you use the in-context editing, you still use a forms-based interface to enter content. Many of OpenText’s competitors now offer a pure wiki-like editing capability, where there is no need to use a forms-based interface.

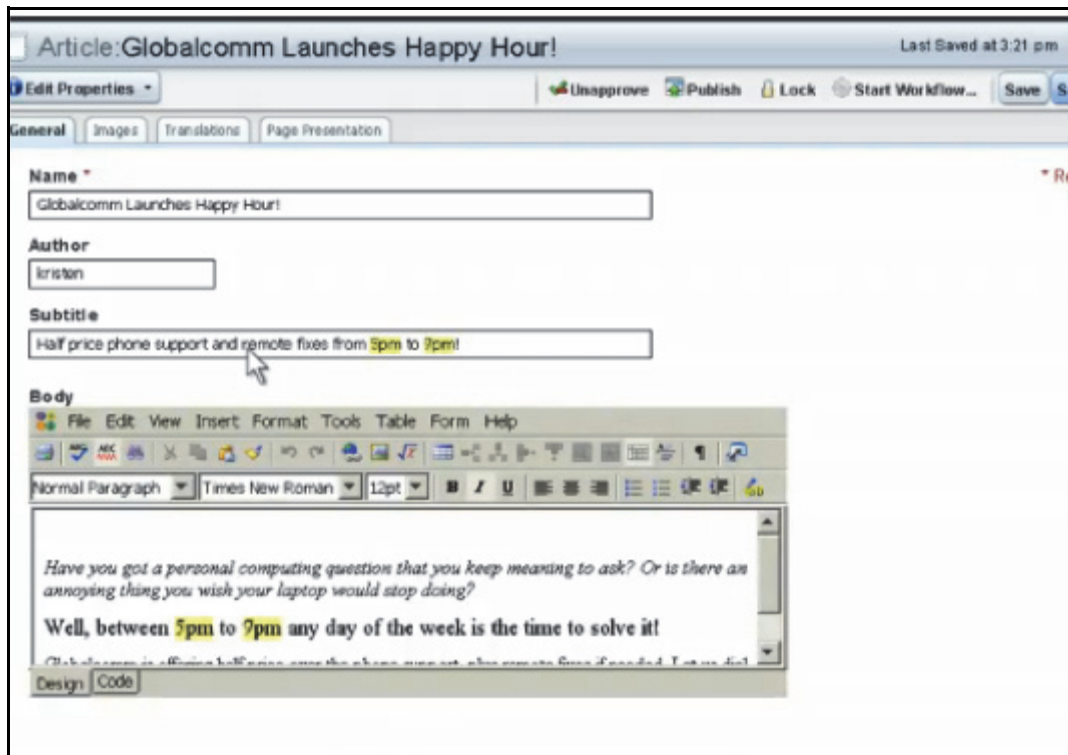


Figure 83. Like many CMS packages, WEM relies heavily on tabs. Ephox's latest EditLive! applet is the WYSIWYG editor.

In addition to browser-based input, WEM boasts direct integration with Microsoft Word. Some WEM customers have used Vignette Business Integration Studio (VBIS) to develop Word-to-HTML conversion routines; however, they must rely on Word “bookmarks” or templates, rather than Word styles the way many other systems work.

Authors can perform in-context editing of page elements (Figure 84) when using the Presentation Management Services for content delivery. This allows tweaking of site contents in full-page preview mode (something most vendors offer these days). Note that with the in-context version, all elements are automatically scoped to either the page or template level. This is mostly helpful, since it prevents casual users from changing items that might appear site-wide — even where authorized. However, it could lead to some misunderstandings if an employee believes that all changes made *will* occur site-wide. More generally, it reinforces the notion that WEM is more page oriented than it is to site-wide component reuse. OpenText claims that WEM can be used in a manner that is not page oriented but more content centric — especially in scenarios when publishing to applications other than the web. You should explore this carefully if you want to use WEM in a manner that is not page centric.



Figure 84. WEM has an in-context editing UI, but faithfully replicating the live site can be tricky.

Authors can query the repository to find and associate related content items, in addition to any dynamic associations (Figure 85). This functionality is particularly helpful for media sites. Note that you are not rearranging or reconfiguring a dynamic list, but simply adding a manual list element.

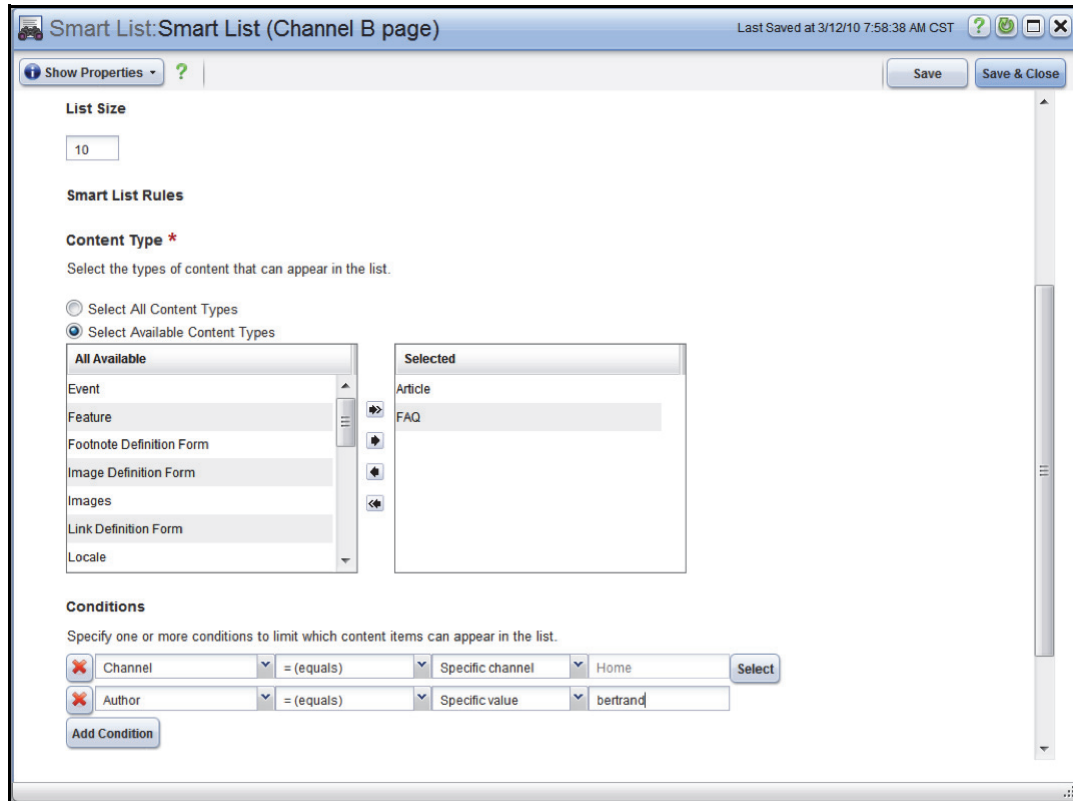


Figure 85. Dynamic queries within specific elements can be modified through a browser interface, but it calls for somebody who is not put off by SQL-like (“find where”) query paradigms.

Although authors can repurpose items by assigning them to different “channels,” granular content reuse remains tricky and requires substantial developer intervention. Renditioning occurs only when content is deployed, so you still need to push content to staging for completely accurate preview, even of static content.

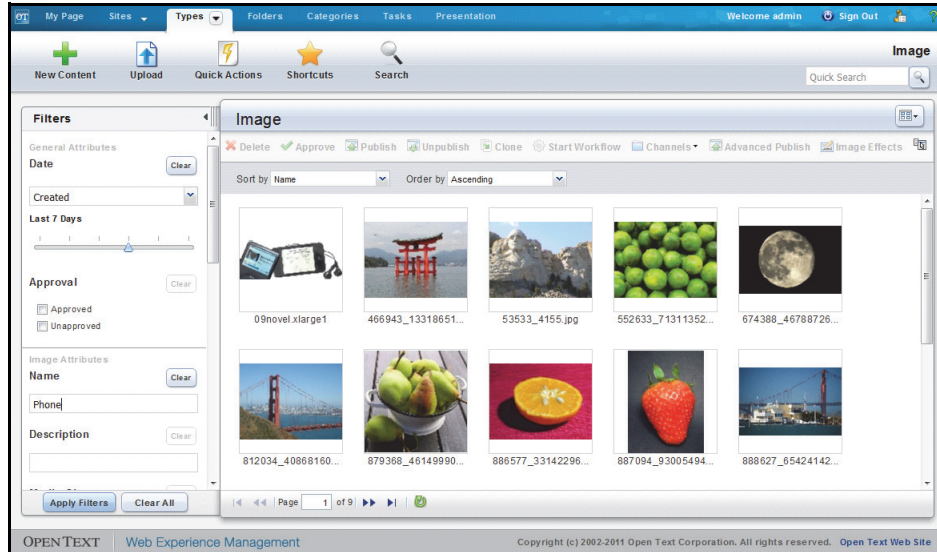


Figure 86. One of the strengths of OpenText WEM is its ability to integrate with the OpenText Media Management product for digital asset management.

Many of these problems become much easier when you use OpenText’s own portal (or Presentation Management Services) for delivery, perhaps another reason why the company pushes it aggressively in conjunction with WEM.

Metadata handling as a whole is somewhat weak compared to some competitors. Although you can apply different types of extended metadata to different content types, OpenText natively tends to fall back on its channel-based classification pattern, which drives navigation (i.e., “Where should this content appear on the website?”).

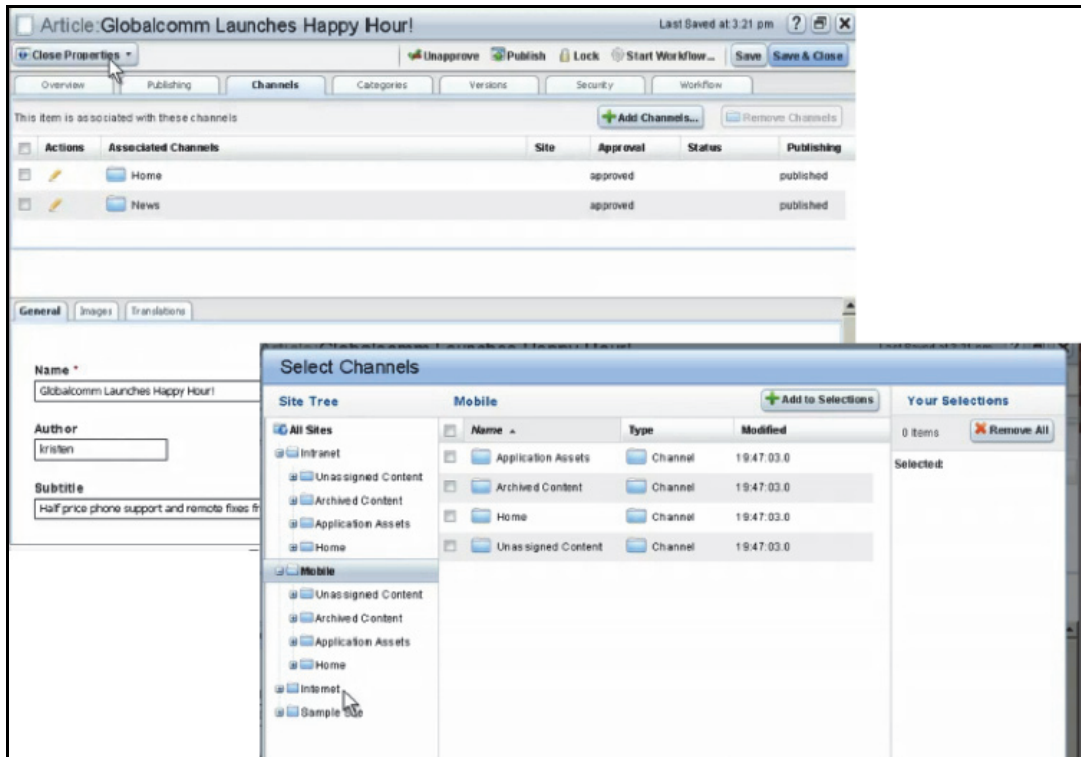


Figure 87. Associating channels to content items has been improved in WEM v8. You now can check the channel names in the same screen by clicking a button.

Contributors can separately classify content via a taxonomy. Using the Categories Workspace, you can traverse the repository based on a Category view. The product can hold only one central taxonomy. A content item is contained within one project, but can have many classification taxonomy associations, and multiple site-channel associations. On the plus side, WEM allows you to source metadata from an external repository — very useful for enterprises trying to standardize on one or more reference repositories. This particular interface has been AJAX enabled, which makes it much easier to navigate, particularly deep or wide trees.

In the end, however, most OpenText customers seem to get confused about the mixture of folders, channels, and classification.

Content Lifecycle

Workflows are aggregated from discrete tasks that you model separately. This may seem non-intuitive at first — especially for those with simple, linear workflows — but this object-oriented approach scales well for firms with many diverse (albeit similar) approval streams. Unlike most other products, you can tie workflows to objects other than just content types, including channels, collections of content, and projects. This means you can have a centralized content type, like “case studies,” but control how it is approved separately in different areas of your organization without creating a new content type. This is an unusual capability that could be quite helpful in super-distributed environments.

In V7, Vignette followed the lead of several other vendors and created a plugin for MS Visio. You model workflows in Visio, and then connect to the CMS to upload the resulting XPDL (XML Process Definition Language) file.

In terms of what the workflow subsystem will support, it's comparatively powerful in that it supports splits, joins, escalation and retries, sub-workflows, and programmatic tasks as first-class activities. It's not just a system for shuffling emails around.

However, end users were originally not happy with the product's workflow usability when it was not exposed through the dashboard. V7's original task list was very generic, and natively, there were few opportunities to annotate tasks and content items as they proceeded to approval. Some of these issues have been addressed in previous versions (which introduced a much friendlier workflow interface), but try it out yourself and evaluate how much customization you'll need to undertake.

New in version 8 is the ability to clone sites. You can create a site template based on an existing site and use that template to create a new site. The new cloned site comes with all the content types, themes, and pages. You can also configure whether you'd like to share content across cloned sites, or replicate content for each of the cloned sites.

Like Oracle, OpenText took some halting — but useful — steps toward real records management in the CMS environment. The company's separate product, OpenText Records and Documents, can manage “external records” — in this case content items in a WEM repository. This sort of federated management is never trivial; in this case, you need to register records as part of a standard workflow and use a special plugin listener to handle declaration. Overall, though, the approach makes sense and feels well conceived.

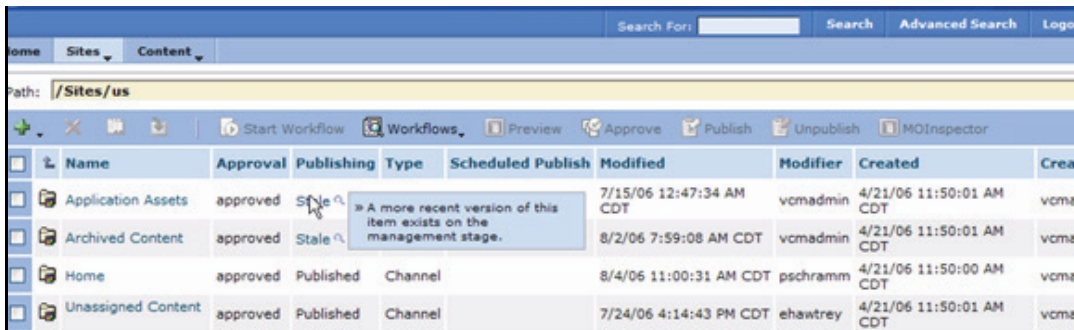


Figure 88. In addition to its optional Records Management software, WEM will indicate when content has gone “stale.”

Here again, OpenText sells an array of its own Records Management products.

Experience

Publishing

Historically, Vignette (like most of its competitors) has created ugly URLs. Thankfully, WEM 7.5 introduced better support for friendly URLs. OpenText V8 has taken this a step further, and now you can have vanity URLs, i.e., you can set URLs for content items, sites, and channels.

Digital Marketing

WEM offers a number of interesting personalization capabilities — if you budget for add-ons. An add-on called OpenText Campaign Management (formally Vignette Dialog) allows interactive, personalized, multi-step and multichannel customizations delivered via email and online. OpenText Web Recommendations (formally Vignette Recommendations), meanwhile, analyzes the online behavior for similar users in order to deliver an enhanced online experience by providing content recommendations, product recommendations, and social search functionality.

OpenText extends into the realm of end-user analytics, including advanced usage reporting and campaign management — but again, this costs extra.

Confusingly, while all this behavioral personalization is based in WEM, profile-based personalization resides within Vignette Portal. OpenText is moving to correct this in the coming years, but it's yet another measure of the Frankenstein architecture (and resulting poor experience for marketers) that you're dealing with here.

Ancillary Services

WEM does not ship with micro-applications (such as wikis or blogs) as part of the core offering. If you want to implement blogs, surveys, and the like within your content management environment, you can integrate with Open Text Social Communities (formerly Vignette Community Services) for surveys, comments, ratings, usage tracking and/or the company's separate Community Applications for blogs, wikis, forums, moderation. The future of these modules remains a bit uncertain, given the plethora of competing tools developed separately by OpenText. We suspect the company will continue to support both sets of technology, though some will win (and lose) out in the longer term. Hedge your bets accordingly.

For search, WEM now natively relies on the OpenText search engine. In earlier versions, you had to use an external search engine (like HP). OpenText will maintain this HP connector if you still prefer to use it instead of their search engine.

Experience	
Publishing	
Standards Adherence	
Multichannel	
Mobile	
Digital Marketing	
Site & Campaign Analytics	
Testing & Optimization	
Segmentation & Personalization	
Social Media Integration	
Promotional Campaigns	
Community & UGC	
Workplace	
Collaboration & Networking	
Dashboard	
Ancillary	
Site Search	
Online Forms	
Module Ecosystem	

Vendor Intangibles

Vignette was never a simple platform, and licensees typically relied heavily on Vignette Professional Services (known as VPS – now Open Text Global Services, or “OTGS”). Customers often reported a love-hate relationship with vendor-provided professional services here. Many customers have come to rely heavily on

OTGS expertise to launch a new application or undertake a difficult upgrade. Some of the same customers will report that OTGS appears unusually disconnected from the company’s engineering force, often taking its own approach and employing its own set of widgets. OTGS is not cheap, and some customers have been surprised after tallying up the final costs.

On the plus side, customers report that tech support is responsive and solves trouble tickets quickly. In an unusually nice touch, customers can register launch dates with OpenText Support and receive faster response as they approach system launch.

Documentation exists, but licensees complain it is quite cursory at times, leaving some features undocumented. A web search will turn up a lot of miscellaneous documentation, much of which is just old enough to be unreliable. Like many companies in this space, OpenText keeps its documentation hidden from public view, giving the impression of a closed, proprietary approach to technology in a time of increasing openness.

This problem becomes magnified, because Vignette did not support as active a user and development community as most competitors do. This has not gotten better since the acquisition. Overall, our impression is that WEM customers don’t feel as connected to OpenText as much as they seem to be with other vendors

OpenText has followed several twists and turns in WEM licensing model, although the latest variant is reasonably straightforward. You can start with price tag of US\$85,000 for a simple information website, which would include the Presentation Management Services as well as Media Management Services. For large and complex deployments the Digital Marketing package that starts at US\$250,000 and includes social, web, portal, and archival solutions. In our experience, most licensees pay much more

In any case, study the product catalog carefully, because features you might assume would be in the core WEM may not be. Start your budgeting at \$250,000 and don’t be surprised if it ramps up sharply from there. By the same token, don’t be shy about negotiating. The WEM group at OpenText needs to stay on new customers’ short lists and land new deals; use this information to negotiate more favorable terms.

While negotiating, scrutinize contractual and financial details carefully. In our experience, Web Experience Management sales demos frequently employ the Portal product and often show optional modules as well. In any case, know what you’re buying, so you can do apples-to-apples comparisons with other offerings.

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

Conclusion

For all its ups and downs, OpenText WEM is still a formidable product in terms of extensibility and scalability, and represents a potentially promising development platform for custom content applications. Across various products, the OpenText WEM platform now offers the social applications, personalization features, advanced caching, video modules, AJAX-powered front-end, and numerous other capabilities that customers have come to expect in offerings at this.

However, you should understand that WEM's value comes at a cost — not just in licensing and support, but also in things like training, maintenance, configuration management, and custom development. Be clear on what you want out of the system and what you're willing to pay. Also, be sure your contributors and editors find WEM's interface acceptable, via careful usability testing with live subjects. The best system in the world can't meet your needs if no one can use it. Perhaps this is why we see more customers leaving this platform than joining it.

Finally, make sure you get comfortable with OpenText's plan for the platform — not just for WEM, but for all of the other modules that typically make up a complete Vignette implementation. OpenText has released a comprehensive roadmap for WEM, but if you are also considering other products, study how they plan to integrate those with the OpenText platform. Most importantly, be very careful to understand all of the different OpenText modules and how they'd really work together in their environment. This company is infamous for its nice-looking diagrams that don't reflect reality once the tools are installed in your datacenter.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Oracle: WebCenter Sites

www.oracle.com

Project at a Glance

Specsheet	Oracle: WebCenter Sites 11gR1
Geography	Global
What's New	<ul style="list-style-type: none"> • Merger of some disparate user interfaces into new Contributor UI and Administrative UI • Enhanced mobile web and portal development tools
Strengths	<ul style="list-style-type: none"> • J2EE-based product that runs on a wide range of operating systems, application servers, and relational databases • A sophisticated development platform for custom content applications • Supports many possible deployment configurations, including tightly, loosely, and decoupled production and delivery environments • Strongly focused on the needs of digital marketers in general and personalization scenarios in particular • Scalability of delivery environment under high loads • Deep LDAP integration can support more centralized security models • Component-based nature of the tool allows for more granular separation of presentation from content, as well as content reuse
Weaknesses	<ul style="list-style-type: none"> • A definitively very complex system to roll out and maintain • Designing new workflows is an arduous process • Multiple contributor UIs adds complexity and is potentially very confusing for line-of-business users • Dual asset model ("basic," "flex") seems like overkill and adds complexity • Customizing the presentation layer or creating new templates typically requires a Java developer • The caching mechanism is very complex, which may hinder performance in incorrect implementations • FatWire/Sites-experienced developers are not as prevalent as larger competitors are • Potentially very pricey under Oracle's CXM Suite umbrella • Web content management is not Oracle's primary focus — and it shows
Potential Fit	Advanced Digital Marketing, Mobile Sites, Global Enterprise
Unlikely Fit	All simpler scenarios, most Digital Workplace scenarios
Compare To	CoreMedia, Adobe, OpenText, SDL
O S	Solaris, AIX, Windows, HP-UX, Linux
Repository	Database: Oracle, MS SQL Server, Sybase, DB2
Client	Browser
App Platform	Java EE application server: Sun, Oracle, Tomcat, IBM, JBoss
Licensing	Lists at US \$80,000 per CPU for delivery server; other servers and modules are extra
Ownership	Public (NASDAQ: ORCL)

Summary

FatWire Corporation of Long Island, New York was one of the larger, privately held, pure-play WCM vendors until Oracle acquired it in July 2011 and subsequently rebranded the offering as WebCenter Sites.

Sites offers a mature, component-oriented, Java-based WCM platform that has grown steadily both in capabilities and sophistication over time. FatWire has traditionally focused on serving the needs of customers in publishing, e-commerce, and catalogs, with a strong emphasis on interactive marketing, and this legacy is still evident. Over the years, the vendor added other modules to its WCM core, including Gadget Server, Community Server, and Mobility Server.

While Oracle’s WebCenter Sites is clearly overkill for simpler scenarios such as an Informational or Basic Digital Marketing sites, it’s worth considering (against the likes of SDL or Adobe) for larger, more complex sites, thanks to a scalable architecture, flexible deployment options. Be aware, though, that you will find comparatively fewer consultant and community resources to help you.

Scenario Fits	
Simpler Site	
Informational	<input type="radio"/>
Microsites & Landing	<input type="radio"/>
Mid-Range	
Basic Digital Marketing	<input type="radio"/>
Mobile Site	<input checked="" type="radio"/>
Community Oriented	<input type="radio"/>
Complex Site	
Advanced Marketing Portal	<input checked="" type="radio"/>
Global Enterprise	<input checked="" type="radio"/>
Multichannel Publishing	<input checked="" type="radio"/>
Ultra-Large Single	<input checked="" type="radio"/>
Digital Workplace	
SMB/Departmental	<input type="radio"/>
Global/Enterprise	<input type="radio"/>

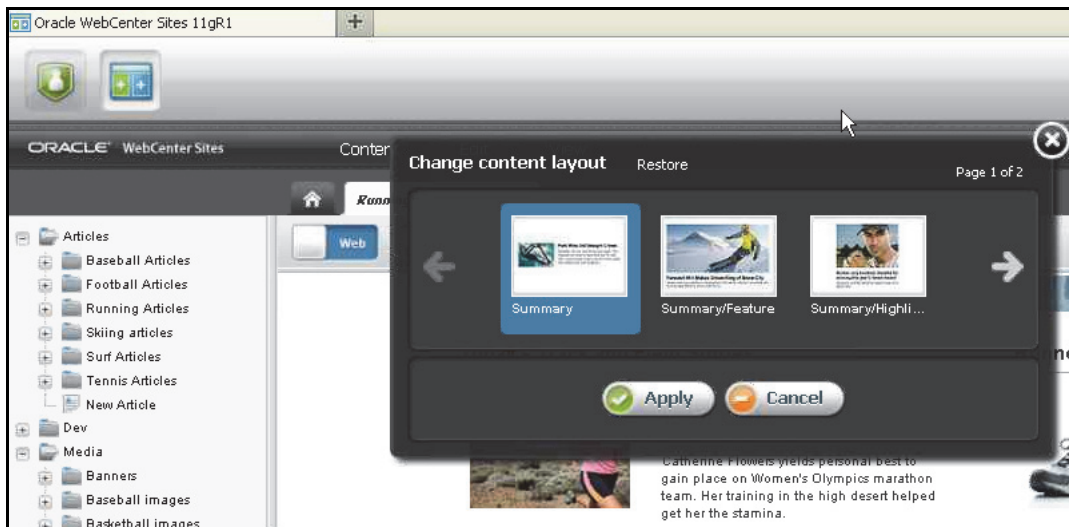


Figure 89. Content authors can choose from predefined layouts/ templates stored and managed in WebCenter Sites.

Finally, you should consider the fact Oracle treats WebCenter Sites as one of the pillars in its gigantic Customer Experience Management suite that is comprised of organic and acquired software; therefore, you always run the risk of overbuying or being oversold. WebCenter Sites is also one of the four components of Oracle WebCenter proposition. Other neighboring products in WebCenter are Portal, Social, and Content (the former UCM offering, now sold for

Enterprise Content Management purposes). Oracle itself cannot consistently articulate the boundaries among these products and encourages you to license them all — for a stratospheric fee — and then sort it out later. Prospective buyers should monitor product bundles carefully — and watch their wallets.

Introduction

Prior to the Oracle deal closure, FatWire was a privately held firm of 200+ employees headquartered in Mineola, New York. The company has hundreds of customers, and slightly more than half of all revenue comes from outside North America.

FatWire acquired the technology for its analytics add-on from 1-2-1 Marketing. In August 2007, the company hired a new CEO: Yogesh Gupta, formerly CTO of Computer Associates. A few months later, FatWire acquired Infostoria, a Setauket, New York, firm specializing in content-sharing technologies.

WebCenter Sites 11gR1 has been available since mid-2012. The release combined two previously separate editorial interfaces into one — known as the Contributor Interface, which also brought some enhancements. The old Advanced UI has been refreshed and is renamed the Administrative UI. As for other details of the release, there was more subtraction than addition, as Oracle tried to integrate Sites with other parts of its CX Suite, and minimize the number of SKUs.

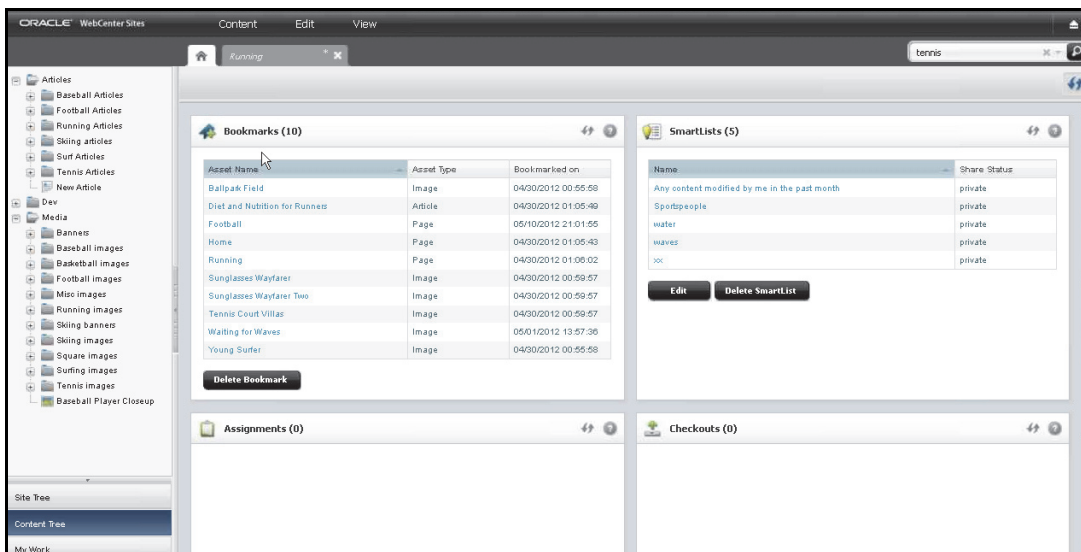


Figure 90. Oracle WebCenter Sites “Home” dashboard showing bookmarks, assignments, checkouts, and other data.

The previously separate module known as Engage (the personalization engine), for example, is no longer sold separately but is offered as part of the core platform.

Technology

Technical Administration & Security

WebCenter Sites has evolved from a coupled management and delivery platform to a system that, today, is typically implemented in multiple tiers (Figure 104), with the final web delivery tier fully decoupled from the rest of the system.

Their “Content Integration Platform” layer traditionally formed the basis for (among other things) connectivity with SharePoint and Documentum via special connectors, but Oracle seems intent on moving this functionality to its WebCenter Content (document management) offering. You’ll also find connectors to Windows and UNIX file systems, and an SDK to build your own. Similar to Adobe Experience Manager’s “Content Bus,” CIP lets you continue storing content right where it lives, while the CMS gives a unified (virtual) view of its management via the WCMS interfaces. Metadata is stored centrally, but files themselves are not. When the system receives a request for a resource, the content is discovered via the central metadata, and then accessed directly from its native store via “agents” that can push or pull content to destinations as needed.

For publishing, WebCenter Sites is a multi-tiered product, typically configured with development, management, delivery, and testing layers (Figure 104). The topology enables flexibility in pushing content out. WebCenter Sites can pre-generate static content, for a “baking” approach, but relatively few licensees take that route exclusively. A big part of what you are buying with WebCenter Sites is a dynamic personalization, caching, and analytics framework. Nevertheless, from a site-design point of view, content can be as dynamic or as fully baked as you want and spread across as many tiers as you need. Be aware, of course, that with CPU-based pricing, you can run your costs up quickly as you scale out across tiers and/or load up WebCenter Sites instances in your delivery environment.

Rather than fully baking your content, you can employ a hybrid “parbaking” system of pre-generating certain elements or even entire XHTML pages for delivery elsewhere. Figure 108 shows deployment options of exporting content to disk (as baked HTML), data, or XML.

The publishing and deployment mechanism is straightforward and simplified to the extent that content is generally published from one WebCenter Sites instance to another. Once a content item has been flagged “Approve Production,” it goes into a queue, where a publishing server (that you may want to install on another machine) generates pages or content snips, and updates all necessary related content. It then pushes those items to the delivery tier and flushes the delivery cache — so far, so good. Unfortunately, while an item is in the queue, it is “locked” and can’t be edited again. Since all publishing systems require a certain amount of wait time to deploy content, if you need to change an item that’s about to go live, you have to either wait or get an administrator to clear the queue. Note that WebCenter Sites publishes content to a delivery database using its own special schema. If you want to use a different

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

delivery engine, or your own custom web application, you'll need to use some sort of ETL process to convert it to a different schema.

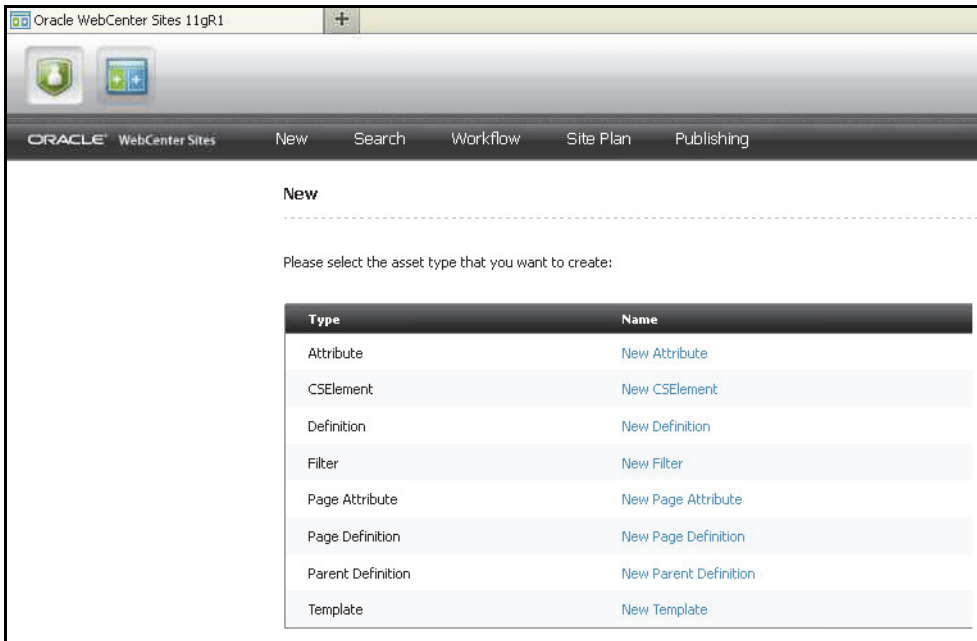


Figure 91. In the administrative (previously called Advanced) interface, you can select what new types of assets you want to create.

On the plus side, you can set up the product to resolve dependencies at content approval time rather than at publishing time, which may be more usable for authors and speedy deployment, even if it might slow workflow. With WebCenter Sites, it is possible to specify deferred publishing jobs, so that a given page (or whole campaign) can go live at a later date and time — and end (go dark) at another date and time.

WebCenter Sites is unusually flexible in its LDAP support, leveraging LDAP not only for authentication but potentially for role management as well. Most Web CMS offerings will allow you to authenticate users against an LDAP directory, and some will let you import users from a directory, with management of users, roles, and site-object ACLs usually taking place inside the CMS itself after users have been imported. WebCenter Sites can be set up this way. However, you also have the option (if your directory is LDAP-3 compliant) of managing users, roles, groups, and site-artifact ACLs in the directory itself. This provides a great deal of flexibility both architecturally and administratively. If the WebCenter Sites administrator has appropriate LDAP administrator rights, the WebCenter Sites admin can make changes using the WebCenter Sites client UI and those changes will propagate through to the directory automatically. If the WebCenter Sites admin does not have LDAP-Czar powers, then they have to work with the admin to set up ACLs and roles. Once everything is set up, the directory admin can continue to “own” all things identity-related within the organization.

The rights model is fine grained. The primary primitives are ACLs, roles, users, and system objects to which ACLs and roles can be associated. ACLs exist as first-class objects independent from other objects; they can be mapped to users and system objects, such that a user can have access to a particular system object (a UI element, for example) only if that element and the user both share the same ACL in common. Roles are flat (not hierarchical)

and serve to delineate system privileges. Many items in the contributor UI are visibility-restricted by role, and roles come into play in workflow entitlements as well. Most ACLs are predefined and you probably won't create new ones. A few predefined roles come with the system, but it's likely you'll want to create custom roles as well. There's no limit on how many roles you can create.

When it comes to controlling access to content, permissions work by setting policies on assets themselves. Within policies, you can define which roles have permissions to perform which functions on assets. For example, a policy might specify which roles are allowed to copy an asset. (This is a surprisingly difficult thing to do in many systems.) For flex asset (see below), policies can be set on parents in such a way that they will be inherited by the children.

Most customers seem to find that the WebCenter Sites rights model more than meets their needs. If anything, you may find it too complex; plan carefully.

WebCenter Sites enables authorized individuals to see what was published and what didn't in any job (Figure 105). WebCenter Sites logs system events and content status, but you can't really query it; it must be sorted by site, user, or time.

Publishing Status

Session ID: 1223529596581
 Status: Failed
 Destination: MySite using RealTime (Complete publish)
 Start Time: 2008-10-30 07:39:46

Failed

Stage	Status	% Completed
Gathering data to publish	Success	100%
Serializing data	Success	100%
Sending data to target	Success	100%
Deserializing and saving	Failed	1%
Updating page caches	Failed	1%

EXCEPTION: Premature end of file. org.xml.sax.SAXParseException: Premature end at org.apache.xerces.parsers.DOMParser.parse(Unknown Source) at org.apache.xerces.jaxp.DocumentBuilderImpl.parse(Unknown Source) at com.thoughtworks.xstream.io.xml.DomDriver.createReader(DomDriver.java:58) at com.thoughtworks.xstream.io.xml.DomDriver.createReader(DomDriver.java:51) at com.fatwire.realtime.SimpleDeserializerImpl.execute(y:2435) at com.fatwire.realtime.DefaultDeserializerImpl.execut

[Configure Destination](#)
[Download Log](#)

Figure 92. The Sites UI includes a real-time monitor, which provides detailed feedback on the publishing process. If a step fails, hovering causes a tool tip to display bearing the Java exception message — which is really only helpful for a developer.

Development

With WebCenter Sites, you typically use (a somewhat complicated mix of) Java classes, JSPs, XML, and XSLT to develop content applications and render pages. WebCenter Sites uses Eclipse-based IDE for developing and previewing artifacts. In the background, the integration creates a file system representation of the assets (JSPs, templates, and metadata) and synchronizes it with the Eclipse environment. This conversion of native assets to managed XML config files is a welcome development, but sometimes a fraught process at first; you'll want to keep an eye out for limitations. For example, it does not initially support workflow assets.

You also can call on several dozen CMS functions that are exposed as WSDL-backed Web Services. As a Java EE system, WebCenter Sites can reach into any database for which a

JDBC driver exists. The Content Integration Platform architecture and associated SDK have gone a long way toward cleanly decoupling the WebCenter Sites management layer from the repository services layer in such a way that integration to other repositories via connectors is now much easier. Application integration is also easier. Thus, WebCenter Sites has become a much better IT citizen than it was earlier in its career.

In terms of templating, WebCenter Sites provides a couple of options for both internal UI and published output. You can modify XML configuration files, which follow a particular WebCenter Sites schema, or for more control, you can create or modify JSP files, optionally adding your own tag library or someone else's. The product's own tag libraries and components are available to page designers via a Dreamweaver plugin, but like Percussion's, you'll still need a developer to create the slots in the first place. On the plus side, you can assign multiple templates to the same content item, and preview it in different contexts. Get used to managing many JSP fragments that call or include each other. As with CoreMedia, OpenText, and Adobe, this system will punish organizations that cannot maintain excellent configuration management.

With respect to content modeling and reuse, Sites took an object-oriented approach further with WebCenter Sites than you would see in the mid-market tier below it, even among the XML-oriented vendors there. WebCenter Sites implements all of its content objects, both text and templates, as content "assets." Historically, a "basic asset" model was available based on an XML "ADF" (asset descriptor file) describing the makeup, user interfaces, and search characteristics of each asset type. Alternatively, you could use a nifty browser-based form to create new asset types.

What is special about WebCenter Sites here is its second asset model, called "flex assets," conceived as an abstraction layer enabling business people to develop more complex inheritance-based content types that might need to span multiple tables in a database. This capability can provide powerful reuse and content-control facilities in the hands of skilled analysts and developers. It was originally developed to compete in the retail product catalog and news-media markets (two traditionally strong areas for the product), but has since been applied to other types of sites with complex content models.

"Asset Processing" APIs enable developers to capture triggers while acting on WebCenter Sites assets and they initiate processes like changing or updating asset metadata, and initiating workflows. This means that developers can now write automated business processes (or batch jobs) that can take runtime decisions — e.g., create thumbnails of images when publishing images and initiate a workflow when one of them fails. This can speed up the business process and potentially reduce the manual interventions required (like eliminating the need to check the log and take appropriate actions). This API also complements Flex Filters; these filters are associated to flex assets and are called to perform post-processing operations when an asset (to which it's associated) is saved.

Some limitations do arise with custom assets that may or may not become an issue. They can be complicated to manage for businesspeople who do not understand the subtleties of content structure. Some degree of training for business users is essential. Flex assets can present a learning curve for developers, as well (particularly when it comes to understanding the underlying SQL data model), and highly nested assets can lead to performance problems when overused — so it's vitally important to understand best practices around flex assets. For best performance, "basic" assets should be used where appropriate (which is to say, in dealing with

“flat” collections of objects that don’t inherit from each other). In short, this is an area to study carefully prior to purchase.

WebCenter Sites makes it simple to glue individual content objects to pages (with some convenient, if rather hidden, dependency reports), but creating automated assembly mechanisms quickly becomes more complicated. In general, mapping objects to pages is complex and will present a significant learning curve for developers. In addition, the auto-generated contributor interface may not match exactly how content editors would like to see it. Customizations can be made, but it requires the understanding of JSP and JSF technology at a code level.

WebCenter Sites is designed to handle multisite management scenarios that involve deployment of sites that might share management resources as well as content resources in many-to-many fashion. Used in conjunction with flex assets and inherited access rights, this means some fairly sophisticated cascading of rights and reuse of resources can occur. A feature called Site Launcher allows you to clone a new site from an existing implementation, including all metadata, templates, workflows, and other configurations. On a content type basis, you can choose whether to copy content or simply share from the single original source.

Many Java API methods are exposed as Web Services, with SOAP bindings and WSDL files automatically pre-configured. As with other vendors who tout “Web Services” APIs, however, results in the field using these capabilities have been mixed. To some extent, this is more a comment on Web Services and SOAP than on WebCenter Sites. The platform supports a REST interface via its WEM framework via an additional layer on top of the CMS and could add additional complexity — along with perceived benefits of interoperability.

As with other platforms at this tier, some customers have been surprised that WebCenter Sites only provides very limited out-of-the-box functionalities, with even simple tasks requiring significant development efforts. Don’t underestimate developer resources required.

Performance

WebCenter Sites ships with a caching engine engineered for high-performance sites. It was initially developed as a separate product for a major European financial newspaper, and then the vendor started shipping it natively with WebCenter Sites. Standard templates in current versions of WebCenter Sites utilize caching. This means that high performance sites can decide to move only the caching server to a different machine to improve performance. In this architecture, the caching server communicates with WebCenter Sites on a separate box. This reduces load on the WebCenter Sites machine and distributes points of failure: If the WebCenter Sites box goes down, the caching server carries on, still with the same targeting and personalization features that a static site cannot deliver. Note, however, that WebCenter Sites charges separate licensing fees for using the caching server on separate machines. Many customers have found that Sites’ caching is rather complex and is one of the areas most prone to costly development mistakes.

Content

Contributor Experience

WebCenter Sites’ main UI is now known as the Contributor UI. Most of the functionalities that were previously dispersed across several distinct interfaces, such as “Dash” and “Insite” have been rolled into one in version 11g. The admin part, however, remains separate. The “Administrative UI” most resembles the previous power-user UI, which bundled a variety of advanced features.

Depending on your role, you may still have to dabble in the Administrative UI, where developers and admins perform advanced tasks. However, content authors will most likely stick to the Contributor UI. While some Sites users may appreciate the new uniformity, others find it cognitively challenging and are still acclimating to the increased bells and whistles they need to navigate through in order to complete their tasks.

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

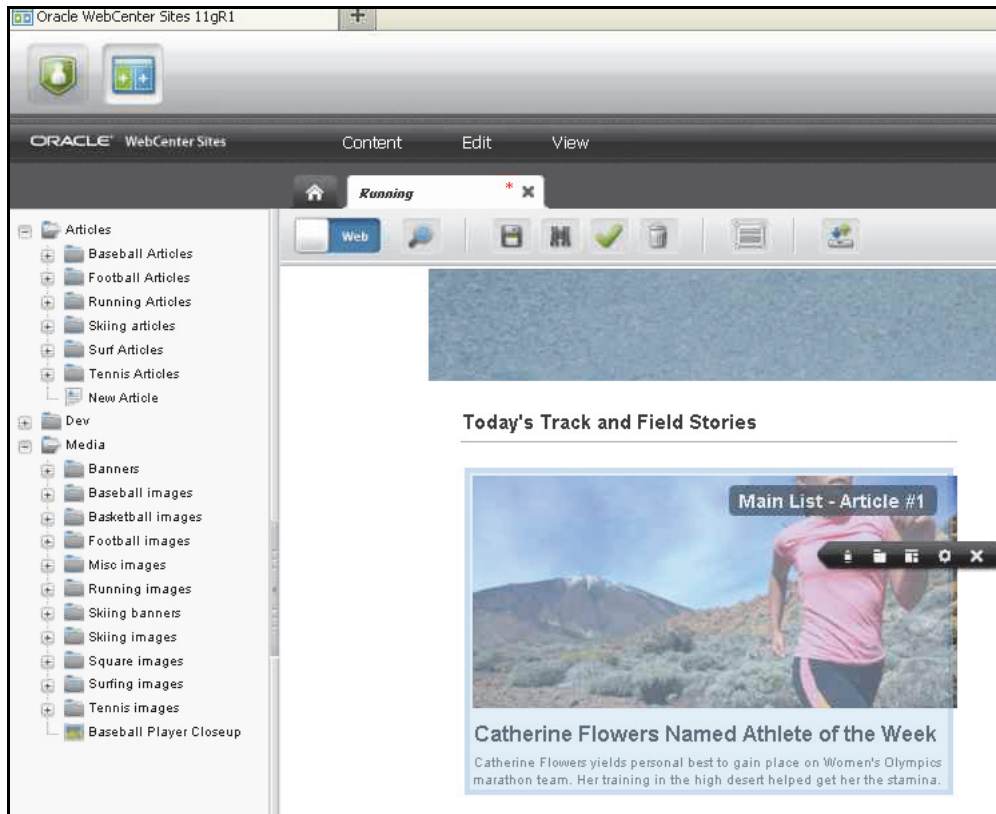


Figure 93. In-context editing in Oracle WebCenter Sites.

You'll also find an in-context editorial environment for non-technical authors, which has seen some enhancements, such as visual search and drag-and-drop functionality.

WebCenter Sites doesn't offer any built-in accessibility checkers or accessibility-related features per se.

WebCenter Sites does acquit itself fairly well in out-of-the-box localization. WebCenter Sites has been localized into French, Italian, German, Spanish, Japanese, Greek, and Portuguese; plus a toolkit exists for delivering additional language versions. A language-picker menu is available in the contributor UI to allow switching between languages on the fly. Standard Java internationalization and localization paradigms apply, in case you want to modify the system.

Contributing Content

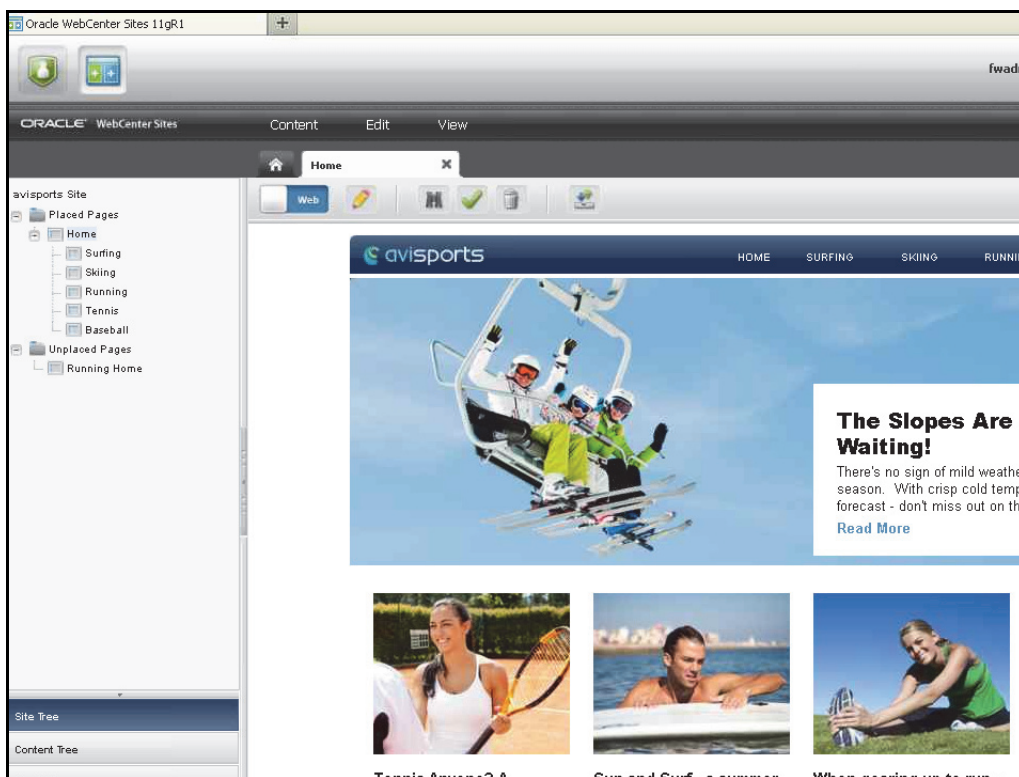


Figure 94. WYSIWYG, visual search, and drag-and-drop functionality govern the latest reincarnation of WebCenter Sites' UI.

The WYSIWYG interface provides access to preview, editing, page builder, search, and workflow functionality. It also employs CKeditor. Be sure to test it if you use non-IE browsers.

WebCenter Sites' in-context editor (like Adobe) gives authors the ability to promote and demote content items within lists on the page. A module called Page Builder enables business users to make changes to the underlying template as well. As always, you'll want to check the quality of the underlying code generated in any WYSIWYG process.

WebCenter Sites can demo useful Word transformation capabilities. In practice, it rarely is used, as it entails persuading your authors to apply styles to their Word document and using a

preset style sheet to boot. If key stakeholders in your project are Word addicts, this may present a problem. Be sure to inform them of the issues around Word usage.

WebCenter Sites' metadata facilities demo nicely, but get a bit creaky behind the scenes. WebCenter Sites has a notion of "folksonomies," where contributors can tag content according to their own categorization scheme and vocabularies. The product only supports one taxonomy tree out of the box, with one set of permissions, although the controlled vocabulary is at least itself a managed object, so you can apply library services and workflow to changes. With a little work, you can create multi-taxonomy structures for managed content with flexible inheritance rules.

Media companies might like some special editorial niceties. Editors can manually re-sort dynamic lists (including related link lists), or even make changes to a particular instance of a dynamic list. This is very handy for supporting editorial judgments on top of metadata-assisted publishing. WebCenter Sites can consume and process XML feeds by means of an XMLPost utility, which Oracle says is used extensively by its customers.

Sites improved the product's document and image management via an optional product called DocLink. You can drag binary assets to and from the repository using Windows Explorer. These assets enjoy the same capabilities as others in the system, including versioning, workflow, and text search, as well as sophisticated asset filtering (e.g., to limit the weight of particular classes of images). Note that, unlike other CMS products that support the WebDAV standard, DocLink requires a client install (Windows only).

WebCenter Sites offers decent image handling, including thumbnail auto-generation and a proper image library. For actual image editing, WebCenter Sites goes well beyond the basics to give you control over image brightness, contrast, hue, and saturation, in addition to a variety of Photoshop-like effects, including embossing, mosaic, oil-painting effects, and so on. Some of these effects strike us as overkill, but conversely, for some graphics-rich environments, where the publishing team doesn't want to continuously round-trip back and forth to Photoshop, this type of approach is practical.



Figure 95. Content authors can drag-and-drop various banners and images from Sites' image library onto pages.

WebCenter Sites also supports Flash animations as first-class media objects, with live preview. Repository search employs an embedded-Lucene solution. You also can configure advanced search to enable search by content type, which can be handy in large-scale systems.

Content Lifecycle

The workflow system follows a state-transition model, based on steps, actions, conditions, and states. Steps can entail more than one possible action and most operations can be restricted based on privilege levels. The system is actually quite powerful and there are detailed configuration dialogs to help you design the steps, states, email objects, and other artifacts that make up a custom workflow.

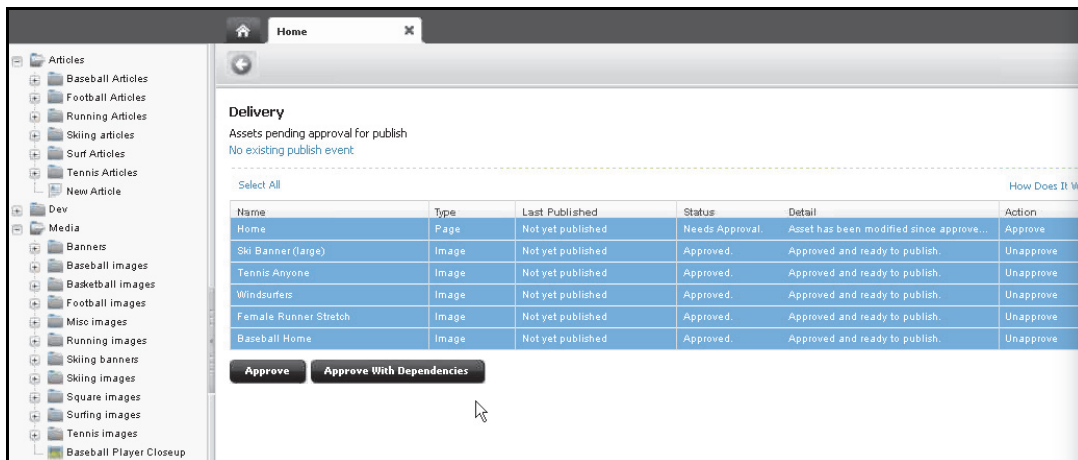


Figure 96. Workflow and approvals in Oracle WebCenter Sites.

However, there is no real flow designer (the vendor suggests that you sketch your design on paper) and the existing dialogs, while helpful as far as they go, don't do much to lighten the burden of designing a workflow — or to keep you out of trouble. (It's easily possible to design flows that permit deadlock.) Of course, any suitably powerful workflow system inevitably brings with it a certain irreducible amount of complexity. Unlike Adobe Experience Manager, WebCenter Sites does little to hide the complexity.

Using flex assets, you can bind multiple content items into collections that you can workflow together, e.g., in a new area of your site — very nice. These functions are all managed from a browser interface, and while these interfaces do not have some more sophisticated features (like stickies or spell check), they are straightforward and complete.

Add New Workflow Process Step

Process Name: Publishing

* Step Name:

* States:

From State	To State
none - (Start of Workflow)	none - (End of Workflow)
BF:Check Criteria	BF:Check Criteria
BF:Fund Parent Completed	BF:Fund Parent Completed
BF:Fund Parent Created	BF:Fund Parent Created
BF:Parent for Approval	BF:Parent for Approval

* Authorized Roles:

- Analyst
- Approver
- ArtworkAuthor
- ArtworkEditor

* Assignment Method:

- Retain "From State" assignees
- No assignments; control actions with function privileges
- Assign from list of participants
- Choose assignees when step is taken
- Assign to Everyone

(Select Roles)

- Analyst
- Approver
- ArtworkAuthor
- ArtworkEditor

Assignment Deadline: Can change
 Use default

Step Actions:

- ApproveForPublish
- NotifyAllParticipants
- SendAssignmentEmail

Step Conditions:

Deadlock Actions:

Voting: All assignees must vote

Workflow Groups: Step is group synchronized

Figure 97. Detailed dialogs help you to create workflow artifacts, but the overall process of designing and implementing a custom workflow is arduous.

Experience

Publishing

URLs created by WebCenter Sites can be idiosyncratic, but the product ships with a URL assembler that, once configured, forms URLs in a path-like manner automatically. Direct editorial control over page names (as they appear in the URL) is also available.

Oracle offers a comparatively broad mobile offering with FatWire’s original Mobility Server, with capabilities for device detection and adaptation based on device capabilities. While it is a separate product, it is integrated with WebCenter Sites via the WEM integration framework and single sign-on.

Remember that it is a PHP-based product with limited capabilities for customization. According to the vendor, this was intentional to keep it a lightweight offering. However, we think managing and maintaining a PHP application with a J2EE application outweighs the advantages of a lightweight application. This remains unchanged under Oracle.

Experience	
Publishing	
Standards Adherence	
Multichannel	
Mobile	
Digital Marketing	
Site & Campaign Analytics	
Testing & Optimization	
Segmentation & Personalization	
Social Media Integration	
Promotional Campaigns	
Community & UGC	
Workplace	
Collaboration & Networking	
Dashboard	
Ancillary	
Site Search	
Online Forms	
Module Ecosystem	

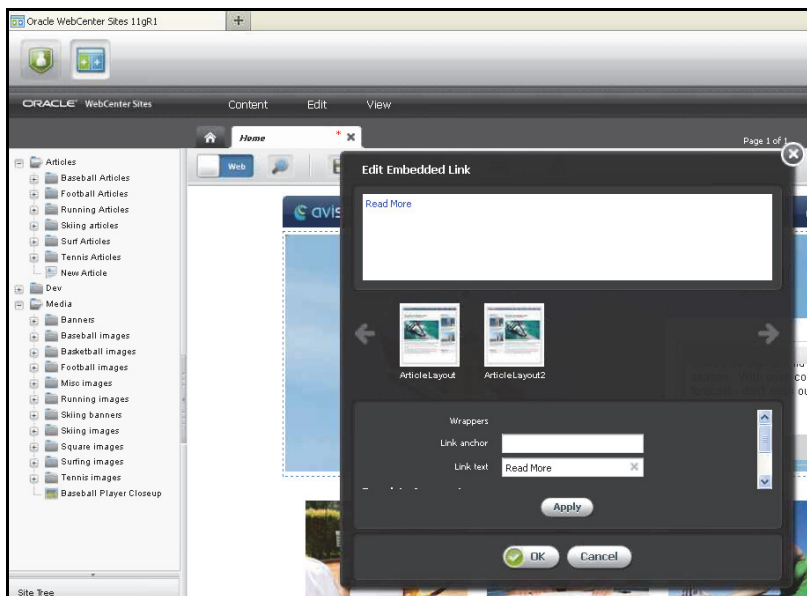


Figure 98. URL and “embedded link” management in Oracle WebCenter Sites.

Digital Marketing

The core product provides profile-based and preferences-based segmentation services. You create profiles, and then assign specific content objects to those profiles. The product also supports the application of some behavioral rules, including runtime recommendations, promotions, and cross-selling, based on relationships established by content contributors. The system sets cookies to track behaviors. Contributors can establish percentage weightings to different attributes to avoid collisions, although this will make application behavior much less predictable and may elongate your testing time. Be sure to test the performance implications, since the more complex your segments, the less amenable to caching they become.

In the latest release, Oracle also introduced targeting automation and optimization services via Oracle’s Real-Time Decisions (RTD) platform, which is their real-time decision and recommendation engine that is loosely integrated with Sites. In the end, you get something dubbed WebCenter Sites Targeting. Oracle is trying to offer a solution to a common customer challenge of not knowing segments in advance and accordingly having difficulty customizing content. With RTD and Sites, Oracle creates segments based on behavioral data (i.e., clicks), and then tries to target and optimize content based on that information.

The role of Sites here is purely on the delivery side, sending clickstream data to RTD. RTD accrues the data, checks the content selected by a marketer, and displays content determined by those rules. Sites receives this call-back and serves up appropriate content to the visitor. While this all sounds good in theory, carefully test the ins and outs of RTD; as of January 2014, it looked and acted like a completely separate application from Sites (similar to SDL’s SmartTarget), with little to no integration among the two — not only from a UI perspective, but with respect to rule-building and monitoring.

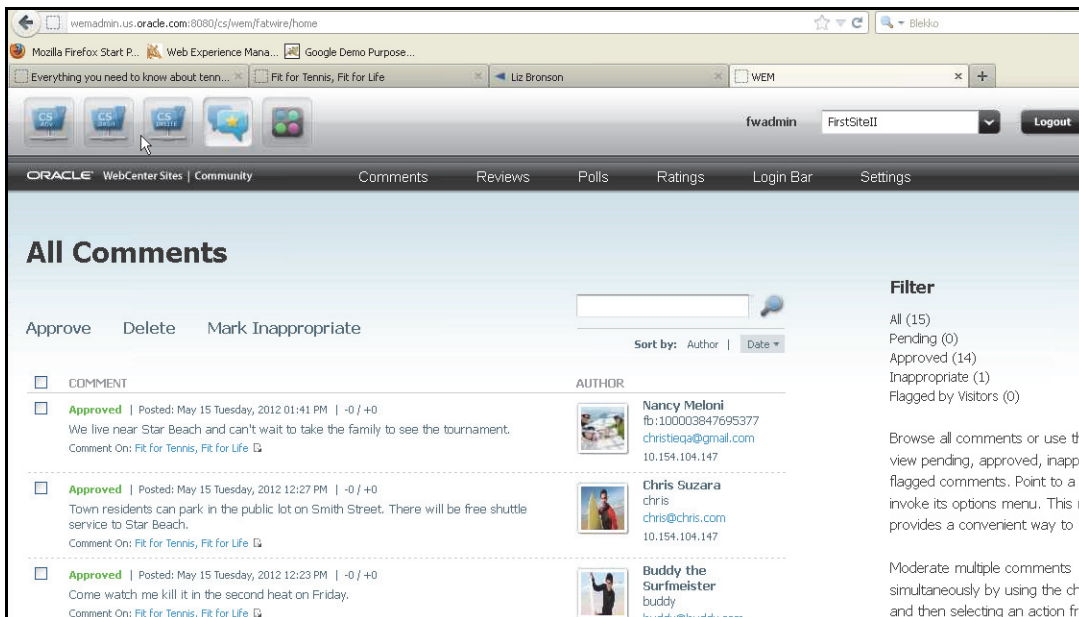


Figure 99. Comments moderation and other community and UGC-oriented features are available in WebCenter Community, a separate package from WebCenter Sites. Do not expect an integrated experience here; even on the UI level, you will be sent to a different interface.

On the analytics side, Sites offers integrations to Google Analytics and similar services, in addition to its own tag-based analytics module. The reason they've developed their own piece of software for this is to use it to feed data back into the personalization settings.

In short, WebCenter Sites is optimized for promoting products and services. Interactive marketers can test different offers and approaches using different content packages and then evaluate how well they do. Optionally, they can fine-tune web page content in iterative fashion to optimize the site. WebCenter Sites does not natively support preferences-based personalization — you would need to code those services separately, or use Oracle's separately licensed Gadget Server.

If there is a shortcoming to Oracle's approach here, it is that the content assembly rules can become rigid and difficult to override manually once in place. Again, the product is optimized for controlled, brand-oriented scenarios.

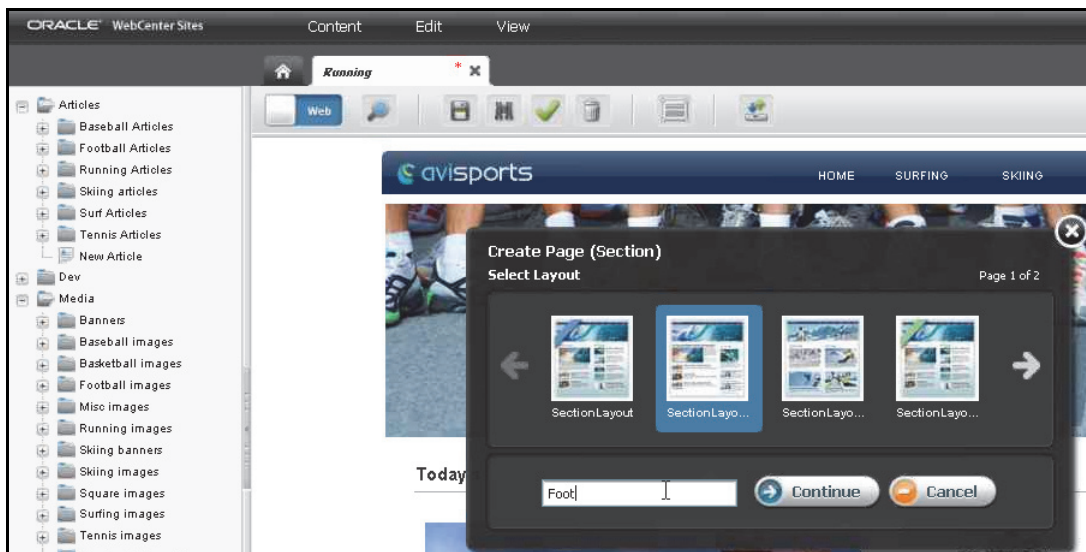


Figure 101. The pop-up shows available templates for creating a new page in this particular context of a WebCenter Sites-managed website.

Of course, dynamic page assembly, personalization, and caching represent the traditional strong points of Java-based portal products. By using WebCenter Sites on its own, you may not need to invest in a Portal product, especially if you don't need to integrate content from disparate sources. Oracle's personalization and caching features may not be as capable as those of some Portal products, but you can manage them within the same interface that you manage your content, typically without developer intercession. This is a plus; only OpenText's Portal (formerly Vignette Portal) and OpenText WEM (formerly VCM) combination, put this together as seamlessly.

However, Oracle will push hard for you to license its own flagship Portal offering, and going forward, has hinted that many of the portal-like features in Sites may be deprecated in favor of WebCenter Portal services. That's a bummer for you.

In regard to UGC, Oracle finally launched polls as a new content type in Sites Community in late 2012. According to the vendor, it was one of the most requested features by the customers. A poll here is essentially a Q-&-A-based form. There's little flexibility with this new content

type and you will have to take it as is. Even a simple modification such as adding a new field to this poll form requires moderate-level customization in Sites, which involves changing the corresponding JSP page and template. The output of the data can be presented as a graph.

Ratings were recently improved by introducing “Improved Styling.” You can tweak it to do thumbs up and down, display “likes,” and recommendations.

Ancillary Services

Sites announced two new offerings in 2010, “Gadget Server” and “Community Server,” both aimed at website visitors. Community Server provides user-generated content services such as blogs, ratings, reviews, and comments. Gadget Server enables you to serve up lightweight components called Gadgets. Both applications run on Oracle’s WebCenter Sites platform. Sites claimed that these products deploy independently, but we are skeptical of the value proposition of this even if it were possible. Both products tie into WebCenter Sites via Oracle’s “WEM” integration framework. This framework enables you to create applications on top of WebCenter Sites and provides features like single sign-on (using the Open source SSO CAS project), unified interface, and a single administration window.

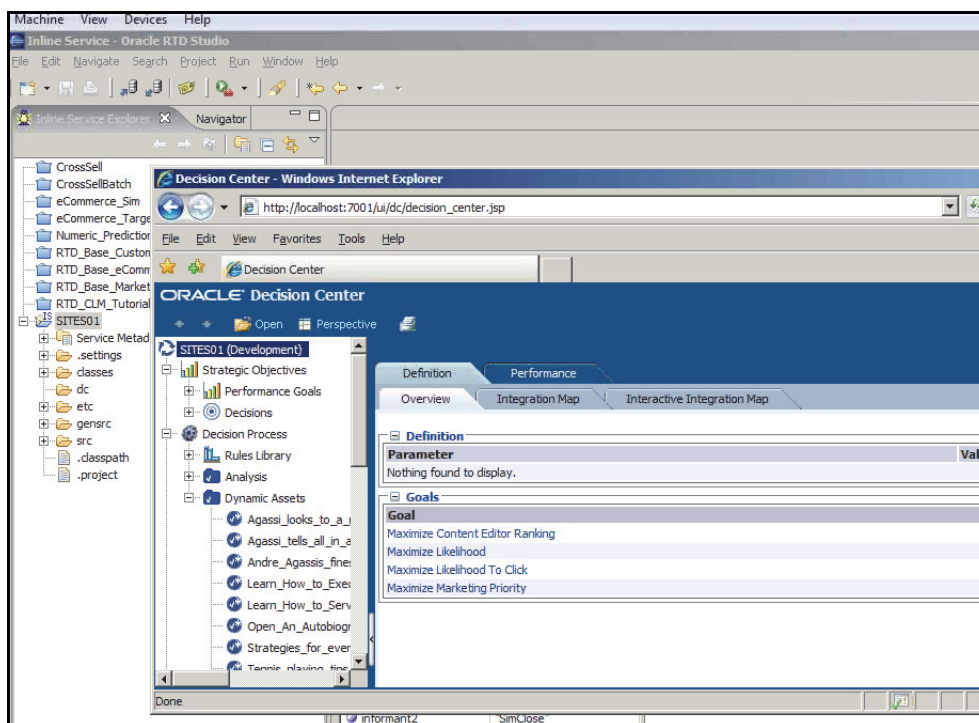


Figure 102. Oracle’s Decision Center is barely integrated with Sites via the WEM integration framework.

The Gadget Server is based on Apache's Shindig project, and you can create gadgets based on Google's Open Social standard. Your site visitors choose the gadgets they want to personalize their interface. Currently, there are only a few gadgets available out-of-the-box, and the percentage of visitors actively modifying page functionality or layout tends to be very low. You may be disappointed to learn that these gadgets are only for visitor-related functionality, and not for editorial functionality, particularly at a time when other competitors employ

gadgets to provide the exact functionality content contributors need. Instead, Sites has only one complex interface that everyone struggles to use.

Overall, the new products provide a useful set of features.

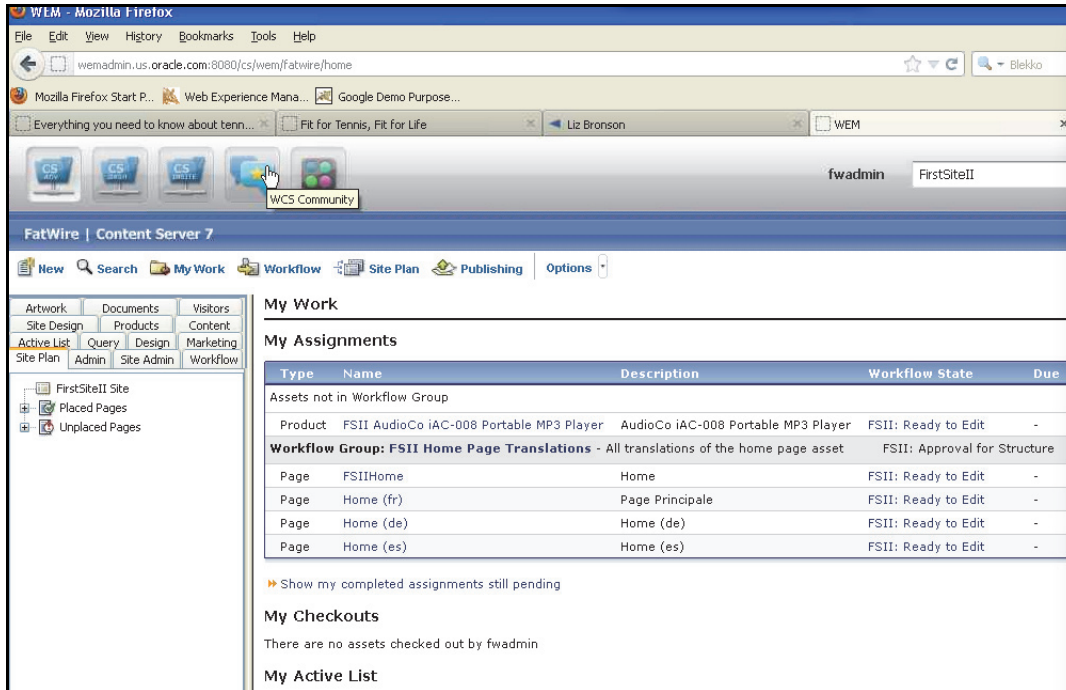


Figure 103. WebCenter Community is, in fact, still the FatWire Community + Content Server disguised inelegantly under the new name. Note FatWire's infamous proliferation of tabs.

With Community, Sites is probably playing catch-up, since many other products have offered this functionality for some time. With Gadget Server, Sites became one of very few vendors to offer in-built gadget services that enable you to run third-party Open Social Gadgets on your infrastructure.

Customers tell us that Gadget Server as well as WebCenter Community were developed in a complete silo and thus are not very well integrated with WebCenter Sites. Review carefully the integration with WebCenter Sites and what Oracle does with these modules before committing to them fully.



Figure 105. Sites’ integration with social media includes Facebook and Twitter logins and sharing.

Vendor Intangibles

With respect to implementations (like some of its competitors), Sites took a partner approach, where the vendor itself keeps quite a few consultants on staff. In general, this is a good way to grow a product, but it also means that in various regions of the world, it can be difficult to find skilled WebCenter Sites specialists.

Oracle continues the former FatWire legacy by balancing its revenues with sizable professional services income, especially in Europe. Given the complexity of the platform, we encourage you to budget more than usual for consulting and support.

WebCenter Sites licensing ranges from US \$40–100,000, depending on a variety of factors, such as the number of processors and additional modules. Satellite Server adds significantly to the basic WebCenter Sites cost, and since pricing is generally on a per-product, per-CPU basis, it’s not hard to run up the bill. Note that previously with WebCenter Sites, you had to pay the same price for development, staging, and production servers, with no difference in price for management versus delivery servers. Now Oracle sells these licenses separately, and the delivery server is the most expensive at over \$100K.

As mentioned earlier however, Oracle doesn’t want to sell you Sites as a freestanding product; they want you to buy into the broader bundle of WebCenter (and related digital marketing) solutions. In that event, you will likely get sucked into an overlapping miasma of products and services that even Oracle itself doesn’t seem to comprehend.

Oracle hosts a developer extranet with a code library contributed by users and partners, but outside of that forum there’s little support or community. The comparatively thin ecosystem

Intangibles	
Vendor Services	
Vendor Professional Services	○
Channel Partner Services	◐
Support & Community	◑
Strategy & Roadmap	◒
Viability & Stability	◓

around FatWire has carried over here, with a customer base having become a bit lost among the huge Oracle community and apps-focused Oracle customer conferences. If you select WebCenter Sites, we hope you become a satisfied licensee, because you will certainly remain a lonely one.

Conclusion

Under Oracle, the former FatWire has buttoned-down and quieted down. After being rolled into the overarching Oracle Fusion Middleware / Oracle WebCenter proposition (along with many other products), Sites has become just a small piece of the puzzle, and is certainly not at the forefront of Oracle's pursuit of a much larger-scale "Customer Experience Cloud."

Pay particular attention to which intra-suite integrations still lie "on the roadmap," and which ones have been "outsourced" to partners (and are, thus far, not truly productized).

Today, Oracle's Web Experience Management offering is best considered within the context of building complex, personalized, content-driven applications. It is most likely to shine when you have catalog-type content in particular.

However, understand that you would be taking on a complicated platform. Despite the considerable amount of work Oracle put into unifying and redesigning the multiple UIs inherited from FatWire, it still suffers from a multi-tab, click-heavy interface that customers often find frustrating. You should test usability here very carefully, and involve your own editorial users in the process.

On the plus side, WebCenter Sites is a relatively mature product, as Java EE offerings go. From an engineering perspective, it stands up well against OpenText WEM, Adobe Experience Manager, SDL, and others in the Java fray. Unfortunately, it costs as much as those platforms as well.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Upper-Range Platform Vendors: Roll-Up Comparisons

- Adobe: AEM Sites
- CoreMedia: CMS
- Percussion: CM System
- SDL: SDL Tridion
- Sitecore: Sitecore CMS

Upper-tier platforms occupy a definitively expanding space above departmental installations, at a time when the “legacy platforms” are largely failing in the marketplace. This is the new “enterprise class.” What that means, however, is that these platforms are typically not as “out of the box” as the Mid-market offerings a tier below. Plan on integration costs ranging from 3-7 times your licensing cost.

On average, license fees run from \$250-500,000; their price points can also rise to higher levels depending on scale and optional modules selected.

Typically, the implementation partners found at this tier run from boutique shops to some of the largest integrators in the world. However, you may find fewer independent consultants conversant in a tool like CoreMedia than you might with Adobe AEM, for example. It simply hasn't been around as long, nor has it been implemented as widely.

Specsheet Summary: Upper-Range Platforms

	Adobe	CoreMedia
Geography	Global	Mostly Europe (particularly Germany), some US, Asia
What's New	<ul style="list-style-type: none"> Name change to AEM "Sites" with version 6.0 Touch-oriented UI is now pervasive 	<ul style="list-style-type: none"> HTML5 editorial interface "CoreMedia Studio" mostly replaces thick Java client Introduction of a CDN architecture in effort to reduce content delivery overhead Early stages of Adaptive Personalization component for creation and management of personalization rules
Strengths	<ul style="list-style-type: none"> Much of the underlying infrastructure is open source and standards-driven, building on a mature content repository Above-average multilingual and multisite management capabilities Flexible content deployment options: coupled, decoupled, or hybrid; as well as cloud deployment options Browser-based visual workflow designer handles most use cases well Large and increasingly active customer community Personalization, targeting, social, and mobile features add value Adobe continues to exhibit strong technical vision and innovation, with interests going beyond core WCXM competency Recent editions have seen improved integration with SiteCatalyst 	<ul style="list-style-type: none"> Mature, scalable product with many original developers still in place Flexible Java architecture Portal-friendly (integrates with many popular portal products) Strong Media industry focus, with experience with large deployments Streamlined in-context editing user interface Strong globalization features Company has a comparatively more deliberate product release cycle
Weaknesses	<ul style="list-style-type: none"> The variety of user interfaces can be unpredictable, awkward, and click heavy, particularly for casual, in-context contributors Adobe appears to have overemphasized the touchscreen management interface to the exclusion of desktop-based experiences Continued uncertainty about how well AEM will mesh with other Adobe products raises lingering concerns with current customers Many functional services (like mobile, social, and DAM) have been moved to pricey add-on products Inspiring technical visions are not always matched by similar quality on the operational and services side; Adobe can be highly disorganized Quite expensive, effectively ruling out AEM for simpler scenarios, since the company focuses increasingly on very large customers Scarcity of partners and independent experts who know how to implement the system adds chronic uncertainty to implementation schedules and costs Lack of module ecosystem forces customers to rely on Adobe for all innovation 	<ul style="list-style-type: none"> Potentially hardware-intensive compared to other systems Combination of CPU and seat-based licensing makes the product comparatively more expensive on larger implementations Doesn't adhere to Java standards as much as some competitors; internal communication relies on dated CORBA protocols Some important integrations (e.g., MS Office, Lotus Notes) have not been productized and remain third-party customizations Some important functions are split between two separate interfaces, potentially confusing users Product is poorly suited for Digital Workplace scenarios Comparatively lighter footprint outside of Germany
Potential Fit	Ultra-Large Single Site, Multichannel Publishing, Advanced Marketing Portal	Global Enterprise, Multichannel Publishing, Ultra-Large Single, Mobile
Unlikely Fit	Informational, Microsites & Landing, Basic Digital Marketing	Informational Site, Basic Digital Marketing, Community Oriented
Compare To	OpenText, Oracle, SDL, Sitecore	Adobe, Oracle, SDL, OpenText, FirstSpirit, Magnolia, CCI

(Continued)	Adobe	CoreMedia
Operating Systems	Windows, Linux, UNIX	Windows/Vista, Linux, Solaris, IBM AIX, HP-UX
Repository	JCR-based repository called CRX (superset of Apache Jackrabbit). Databases: Oracle, Sybase, Informix, MS SQL Server, DB2	Database: Oracle, DB/2, Microsoft SQL Server, PostgreSQL, MySQL5
Client	Any browser (with some IE8 limitations)	IE and Firefox are the only supported browsers
App Platform	Java EE Runs on Tomcat or major Java appservers	Java: Can be deployed to a major J2EE app server or servlet container
Licensing	Base (without modules) starts at US \$120,000 for one authoring server plus one production server. Additional server instances \$50K each. Optional modules are around \$50K each, per instance.	CPU based at \$28,000 per core, plus per-user fee on a sliding scale
Ownership	Public (NASDAQ:ADBE)	Privately held, VC financed — largely held by founders

	Percussion	SDL	Sitecore
Geography	Global, with emphasis on North America and the UK	Global, with emphasis on Europe & North America	Global, but strongest in Europe and North America
What's New	<ul style="list-style-type: none"> • Introduction of an additional product called CM1 as a simpler alternative to its traditional CM offering 	<ul style="list-style-type: none"> • Company released new edition, with somewhat refreshed UIs tailored to user groups, and advanced workflow services 	<ul style="list-style-type: none"> • Changes in the licensing model combines DMS (Digital Marketing System) with core CMS • Multiple enhancements to the search functionality, such as boosting, tagging, and facets
Strengths	<ul style="list-style-type: none"> • Mature Java framework is developer friendly, with extensive APIs, and decent standards support • Very strong native support for component content reuse, and CMS user interface can be set up to follow accessibility standards • "Active Assembly" can give power users substantial control over page development and layout • Decoupled architecture is a very good fit for many enterprise scenarios with custom or heterogeneous website delivery applications • Percussion is a privately-owned, pure-play WCM vendor with no sideline businesses or "parent corporation" to distract them • CM1 product provides an alternative for simpler scenarios 	<ul style="list-style-type: none"> • Comprehensive support for XML and "Schema embedding" can facilitate comparatively good content reuse and re-purposing • Flexible workflow system supports complex approval processes • Well suited to multinational firms with substantial globalization needs and/or mixed Java/.NET delivery tier environments • Loosely coupled architecture may suit some enterprise infrastructures • Blueprinting functionality provides very good support for multisite management in highly distributed environments • Unusually good support for full-cycle email newsletter publishing • Developer extranet and regional user groups are well regarded 	<ul style="list-style-type: none"> • Product is highly extensible and componentized, and can offer portal-like functionality akin to SharePoint • Strong localization bent useful for multinational environments • Solid Lucene integration enables advanced search functionality • Best suited for highly structured content and scenarios with substantial integration with other internal systems, especially Microsoft-based • Feature-rich, Windows-like interface will appeal to some end users • Company boasts a growing stable of third-party resellers and integrators around the world

(Continued)	Percussion	SDL	Sitecore
Weaknesses	<ul style="list-style-type: none"> This is a tech-heavy platform: most implementations will require custom Java development; templating requires a developer Admin interface has always been rather clunky, and the contributor interface tends to cater to power users Personalization, analytics, and UGC features bring ample added costs Workflow system can feel rigid and unnecessarily complicated Customers have said that sales, support, and training can be erratic Third-party specialists and integrators with Percussion experience can be comparatively very hard to find Company traditionally has tended to favor power-user features and architectural sophistication over ease of use CM1 is a separate product that could distract or divide company efforts Company has experienced some leadership turnover, which may bode poorly in future performance 	<ul style="list-style-type: none"> Complete solution may require multiple additional licenses above the introductory package; be careful of sandbagged initial bids Entitlements subsystem is surprisingly underpowered for the kind of global, distributed operations the company targets Kaleidoscope of different technologies (COM, Java, .NET) under the hood adds substantial complexity to development and maintenance Experienced Tridion specialists are globally in relatively short supply User interface is comparatively complex and difficult to use, despite recent redesigns and refreshes Company emphasizes functional development over (long-postponed) underlying architectural improvements Consistently lags behind major competitors in digital marketing 	<ul style="list-style-type: none"> Comparatively developer-centric: More of a platform than an out-of-the-box solution, with longer implementation times Older versions do not behave well in Firefox; best performance is in non-IE browser Chrome Workflow is more developer friendly than user friendly Overly rich interfaces may confuse casual contributors Multisite management services are weaker than direct competition Comparatively poor taxonomy functionality may require workarounds Product typically purchased through a third-party reseller or integrator, leaving the customer once removed from the vendor
Potential Fit	Ultra-Large Single Site, Global Enterprise, Multichannel Publishing	Multichannel Publishing, Global Enterprise	Global Enterprise, Multichannel Publishing, Advanced Marketing Portal
Unlikely Fit	All simpler scenarios, Advanced Marketing Portal	Simpler and Mid-Range Scenarios, most Digital Workplace Scenarios	Informational, Basic Digital Marketing
Compare To	HP, SDL, Oracle, Adobe, OpenText WEM	HP, Adobe, Oracle, Sitecore, OpenText	Ektron, Microsoft SharePoint, SDL, EPiServer, Adobe
Operating Systems	Windows, Solaris, Red Hat Linux	Windows, Solaris, IBM-AIX, HP-UX, and Red Hat Linux	Window Server
Repository	Databases: Oracle, MS SQL Server, IBM DB2	Database: Oracle, SQL Server, IBM DB2	Databases: Oracle, MS SQL Server
Client	All major browsers on Windows, Firefox on Mac. Administrative UI uses applet technology, as does the rich text editor	IE, Firefox, Chrome, and Safari	Browser: IE preferred; Firefox, Chrome, Safari possible
App Platform	Java/J2EE — bundled with JBoss	COM & .NET in management tier, Java/.NET in delivery tier	.NET 4.5; Windows Azure, ASP.NET MVC

(Continued)	Percussion	SDL	Sitecore
Licensing	Server based. Median deal size roughly US\$150,000	Server, CPU, user, and module based, with typical deals starting at \$150,000–250,000+	Based on servers and users, with a variety of modules and packages. Median deal size is \$200K, and can go up from there
Ownership	Privately held	Public (LSE: SDL)	Privately held

Category Summary: Upper-Range Platforms

Phase / Attribute	Adobe	CoreMedia	Percussion	SDL	Sitecore
Technology					
Technical Administration & Security					
Threat Prevalence					
Authentication & Authorization					
System Reporting					
Multisite Management					
Cloud Services					
Development					
Configuration & Customization					
Integration & Extension					
Content Modeling					
Templating					
Performance					
Back-end Performance					
Site Caching & Delivery					
Content					
Contributor Experience					
Overall Usability					
UI Accessibility					
Contributing Content					
Authoring & Transformation					
Tagging & Taxonomy					
Content Reuse					
Media & Document Management					
Repository Services					
Content Lifecycle					
Workflow					
Globalization					
Archiving & Compliance					
Experience					
Publishing					
Standards Adherence					
Multichannel					
Mobile					

Phase / Attribute	Adobe	CoreMedia	Percussion	SDL	Sitecore
Digital Marketing					
Site & Campaign Analytics	●	○	●	●	●
Testing & Optimization	●	○	○	○	●
Segmentation & Personalization	●	●	●	●	●
Social Media Integration	●	●	●	●	●
Promotional Campaigns	●	○	○	●	●
Community & UCG	●	●	●	●	●
Workplace					
Collaboration & Networking	○	○	○	○	●
Dashboard	●	○	○	●	●
Ancillary					
Site Search	●	●	●	●	●
Online Forms	●	●	●	●	●
Module Ecosystem	●	○	○	●	●
Vendor Intangibles					
Vendor Services					
Vendor Professional Services	●	●	●	●	○
Channel Partner Services	●	●	●	●	●
Support & Community	●	●	●	●	●
Strategy & Roadmap	●	●	●	●	●
Viability & Stability	●	●	●	●	●

Key	
○	Product does not provide this feature in any meaningful way.
◐	Product provides this feature, but is not as mature as its rivals
◑	Product provides this feature
◒	Product excels at this feature, <i>relative to other products in the same category</i>
◓	Product masters this feature, <i>relative to other products in the same category</i>

Scenario Fits Summary: Upper-Range Platforms

Phase / Attribute	Adobe	CoreMedia	Percussion	SDL	Sitecore
Simpler Site					
Informational	○	○	○	○	○
Microsites & Landing	○	◐	◐	◑	◑
Mid-Range					
Basic Digital Marketing	○	○	○	○	◐
Mobile Site	◐	◐	◐	◐	◐
Community Oriented	◐	○	○	○	◐
Complex Site					
Advanced Marketing Portal	◐	◐	○	◐	◑
Global Enterprise	◐	◐	◐	◑	◑
Multichannel Publishing	◑	◑	◑	◑	◑
Ultra-Large Single	◑	◑	◑	◐	◐
Digital Workplace					
SMB/Departmental	○	○	○	○	◐
Global/Enterprise	◐	○	◐	○	◑

Key	
○	Product does not provide this feature in any meaningful way.
◐	Product provides this feature, but is not as mature as its rivals
◑	Product provides this feature
◒	Product excels at this feature, <i>relative to other products in the same category</i>
●	Product masters this feature, <i>relative to other products in the same category</i>

Adobe: AEM Sites

www.adobe.com

Vendor at a Glance

Specsheet	Adobe: AEM Sites 6.0
Geography	Global
What's New	<ul style="list-style-type: none"> • Name change to AEM "Sites" with version 6.0 • Touch-oriented UI is now pervasive
Strengths	<ul style="list-style-type: none"> • Much of the underlying infrastructure is open source and standards-driven, building on a mature content repository • Above-average multilingual and multisite management capabilities • Flexible content deployment options: coupled, decoupled, or hybrid; as well as cloud deployment options • Browser-based visual workflow designer handles most use cases well • Large and increasingly active customer community • Personalization, targeting, social, and mobile features add value • Adobe continues to exhibit strong technical vision and innovation, with interests going beyond core WCXM competency • Recent editions have seen improved integration with SiteCatalyst
Weaknesses	<ul style="list-style-type: none"> • The variety of user interfaces can be unpredictable, awkward, and click heavy, particularly for casual, in-context contributors • Adobe appears to have overemphasized the touchscreen management interface to the exclusion of desktop-based experiences • Continued uncertainty about how well AEM will mesh with other Adobe products raises lingering concerns with current customers • Many functional services (like mobile, social, and DAM) have been moved to pricey add-on products • Inspiring technical visions are not always matched by similar quality on the operational and services side; Adobe can be highly disorganized • Quite expensive, effectively ruling out AEM for simpler scenarios, since the company focuses increasingly on very large customers • Scarcity of partners and independent experts who know how to implement the system adds chronic uncertainty to implementation schedules and costs • Lack of module ecosystem forces customers to rely on Adobe for all innovation
Potential Fit	Ultra-Large Single Site, Multichannel Publishing, Advanced Marketing Portal
Unlikely Fit	Informational, Microsites & Landing, Basic Digital Marketing
Compare To	OpenText, Oracle, SDL, Sitecore
OS	Windows, Linux, UNIX
Repository	JCR-based repository called CRX (superset of Apache Jackrabbit). Databases: Oracle, Sybase, Informix, MS SQL Server, DB2
Client	Any browser (with some IE8 limitations)

Specsheet	Adobe: AEM Sites 6.0
Platform	Java EE Runs on Tomcat or major Java appservers
Licensing	Base (without modules) starts at US \$120,000 for one authoring server plus one production server. Additional server instances \$50K each. Optional modules are around \$50K each, per instance.
Ownership	Public (NASDAQ:ADBE)

Summary

The WCM system formerly known as CQ — originally Day Communicé — became part of the Adobe software conglomerate via the 2010 acquisition of Switzerland-native Day Software. Under Adobe, the product is being positioned as part of the overarching Adobe Marketing Cloud and has been through a slew of rebrandings throughout its history: from Adobe “Web Experience Management” to Adobe Experience Manger (AEM). In its newest incarnation, the WCM platform is called AEM “Sites,” to distinguish itself from other AEM offerings.

Adobe has been focusing on integrating AEM Sites with 20+ siblings of the larger Adobe family — starting with SiteCatalyst and Test&Target personalization and targeting products, with somewhat mixed results.

Sites is an architecturally elegant Web CMS aimed at Global 2000 customers that — for governance or other reasons — require a rigorously standards-driven approach to content management, with active clustering, hot failover, decent workflow, developer-centric extensibility, and flexible delivery and deployment options. However, despite its infrastructural elegance, AEM business users seem relatively unimpressed, beyond a few snazzy drag-and-drop capabilities.

As you might expect of anything engineered in multilingual Switzerland, AEM is particularly rich in internationalization and localization-oriented capabilities of the kind needed to build and manage multiple sites in multiple regions, with functionalities designed to enable international workflow, automatic use of localized text and images coupled to different presentation skins, and a localizable user interface for content contributors, among other capabilities. As we’ll see in this report, however, it lacks some capabilities with respect to managing multiple global properties on a long-term basis.

While AEM is overkill for simpler and most mid-range scenarios, it’s a potential match for larger enterprises with unusually complex or diverse needs. Bear in mind, though, that an Adobe solution will be anything but inexpensive, even before you start adding extra modules (e.g., “Social Communities,” “Multi Site Management,” and “Apps”).

Scenario Fits	
Simpler Site	
Informational	<input type="radio"/>
Microsites & Landing	<input type="radio"/>
Mid-Range	
Basic Digital Marketing	<input type="radio"/>
Mobile Site	<input type="radio"/>
Community Oriented	<input type="radio"/>
Complex Site	
Advanced Marketing Portal	<input type="radio"/>
Global Enterprise	<input type="radio"/>
Multichannel Publishing	<input type="radio"/>
Ultra-Large Single	<input type="radio"/>
Digital Workplace	
SMB/Departmental	<input type="radio"/>
Global/Enterprise	<input type="radio"/>

Given a relative scarcity of implementation talent, you will also pay dearly for outside help. This is a high-end system with a high TCO to match. On the other hand, if you're prepared to do a fair amount of custom development and require a WCM platform with a highly extensible architecture and rich set of potential capabilities, AEM could be a good match. As always, weigh your requirements carefully — and be honest about your ability to deal with the complexities and costs of Java EE middleware. In addition, remember that Adobe has not yet proven that the vendor and its partners can consistently deliver the necessary expertise to the complex projects this tool seems to attract.

Introduction

When it purchased Day Software in 2010, Adobe inherited a CMS vendor that competed internationally: targeting large, multinational companies that need to manage dozens or hundreds of geographically dispersed sites serving millions of page views per day. AEM Sites is a complex web content and experience management platform with a multitude of add-ons that can do different things from social collaboration to web analytics integrations.

The company is proud of the fact that its own people have driven some important industry standards:

- Day CTO and cofounder David Nuescheler — now VP of Enterprise Technology at Adobe — spearheaded JSR 170, and was the spec lead on version 2.0 of that specification, better known as JSR 283 (or simply JCR, the Java Content Repositories)
- Day's Chief Scientist, Roy T. Fielding, is one of the original authors of the HTTP specification and is widely regarded as the father of “REST” (Representational State Transfer, a client-server interaction paradigm)
- Some of the most active committers on the Apache Jackrabbit and Sling projects are Day employees, now working for Adobe

Day was also involved in Chemistry, an Apache-incubated project aimed at creating a Java, Python and PHP reference implementation of the CMIS specification. (CMIS, or Content Management Interoperability Services, is an OASIS-ratified standard for interacting with ECM repositories.)

Notably, Communiqué was the first CMS to leverage JCR (Java Content Repository) technology — which should not come as a surprise, considering that Day personnel have played (and continue to play) a seminal role in the development of that technology.

From a licensing perspective, Adobe tweaks its licensing model every now and then — sometimes simplifying it for customers and prospects to understand, and sometimes achieving the exact opposite, resulting in customers' utter confusion.

This is particularly the case with Sites, since it has additional lighter-weight capabilities that exist (in a deeper way) in some of its sibling AEM products, plus Sites has some optional modules of its own. Adobe demos seem to assume you have licensed all of this and more, so be careful about what you think you're buying.

In addition to Sites, other notable AEM products include:

- **Marketing Campaign Management (MCM)**, for marketing automation activities
- **Multi Site Manager (MSM)**, an add-on tool for managing multilingual and multi-brand websites via blueprinting functionality

- **Individual connectors** to other repositories, including those of IBM/FileNet P8 Content Manager and Lotus Notes, Microsoft SharePoint, and OpenText Livelink, as well as connectors to marketing platforms like ExactTarget and Salesforce
- **Social Communities**, for white-label community and social services on your site
- **Assets**, the former Adobe DAM offering (reviewed in our [Digital & Media Asset Management](#) report), which has several optional modules of its own
- **Apps**, a mobile experience management platform that’s a commercial version of Apache Cordova (a.k.a., PhoneGap Enterprise), reviewed in RSG’s [Enterprise Mobile Technology](#) report
- **Forms**, for managing and deploying forms, which is now broken out as a separate product

Beyond these tools, you can license a variety of workflow, analytics, and creative tools from Adobe.

From a business perspective, the biggest change is a new touchscreen UI in AEM version 6.0. Under the hood, however, developers and administrators will see several changes, outlined below.

Technology

Technical Administration and Security

Out of the box, AEM is a Java application that uses Adobe’s own CRX (Content Repository eXtreme) as the repository, Apache Felix (the OSGi implementation) as a services container, and Java Server Pages (or a new templating framework called “Sightly”) exposed through a custom servlet engine — plus application logic implemented in custom Java Servlets and OSGi Services. Note that you can substitute COTS products for many of the open source pieces. Many customers use Tomcat as the servlet container, and Adobe publishes detailed instructions for installing AEM into JBoss, WebSphere, and WebLogic appservers.

Abstraction of the storage layer is one of the prime motivations behind JCR: Applications don’t have to know or care about the underlying storage. Adobe offers connectors to Documentum, SharePoint, and other popular repositories, so that in many cases, organizations can keep their physical data exactly where it is, and still use AEM to provide a repository view into the content, with versioning and other services provided via CRX. This is a major selling point for organizations that have made a significant prior investment in other repositories but who want a WCM system to manage it all.

The CRX repository is a superset of Apache Jackrabbit. Whereas Jackrabbit is strictly a community effort, CRX is a highly productized repository that adds professional administration tools, repository diagnostics, special package-import and export tools, shared-

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

nothing clustering, point-in-time system snapshots, JCR node-type administration tools (essential for custom content modeling), and other capabilities. Also, as a commercial product, CRX features certified builds and service-level support options.

AEM version 6 employs the newest version of CRX — based on the latest version of Jackrabbit called “Oak,” which offers new persistence options (including NoSQL datastores), somewhat faster performance, and Solr-based search.

A major architectural enhancement that differentiates AEM and CRX from earlier releases is its use of OSGi as the runtime resource management infrastructure. The implementation used here is that of Apache Felix — which happens to have its own very detailed administration console.

The downside here is that customers report that the OSGi framework embodies a very steep learning curve. That accounts for some (but certainly not all) of the challenges that even Java shops encounter when trying to learn AEM.

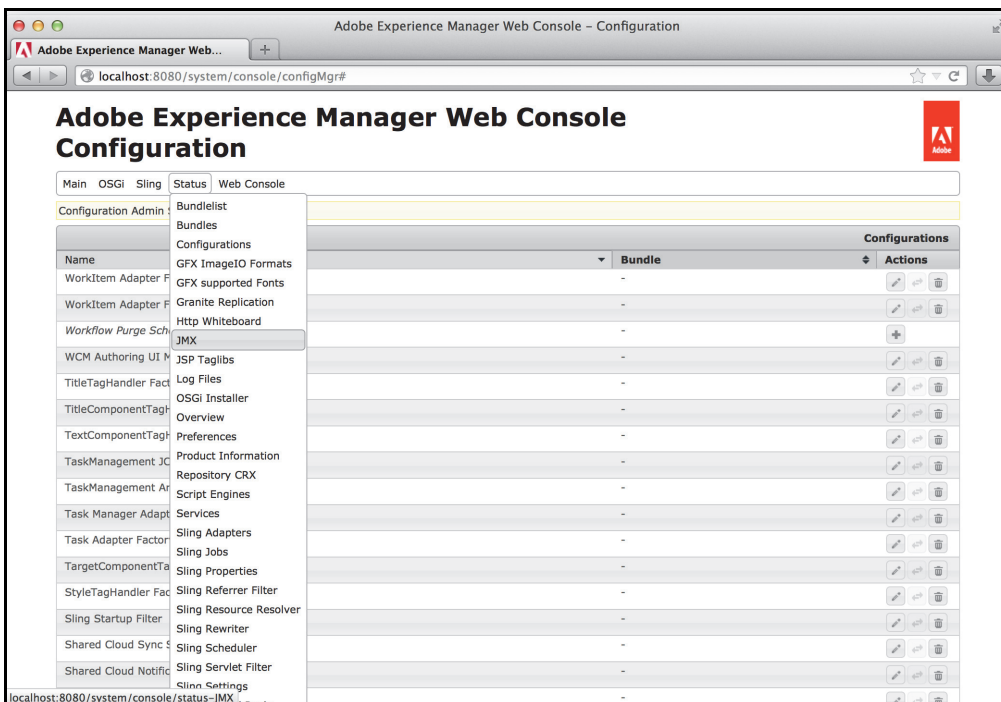


Figure 106. The main console of CRX highlights a myriad of available repository administration tools.

AEM is typically set up to deploy content and code to a web server (such as Apache), Java application server (such as JBoss), or servlet container (e.g., Tomcat) as desired, using standard J2EE packaging mechanisms. For deployment, Adobe uses delivery and receiver agents that communicate via HTTP, FTP, or JMS. In general, it’s a flexible approach that offers transactions and rollback. Using HTTP, the delivery agent usually “pushes” content, but you can set up a “pull” system, as well. Many physical configurations are possible, and the vendor has published guidance for some of them.

In a fully decoupled delivery layout, two instances of CRX (one inside the firewall and another in the delivery environment) can be set up in such a way that publishing across the firewall is uni-directional, with no backflow into the management environment. In a coupled layout, the

same repository can be used for management as well as for delivery. However, AEM also offers a hybrid option in which content is baked out to the delivery tier as needed, and re-baked on demand as new material is published.

With demanding availability and performance requirements, you should explore Adobe's CRX clustering options. Read-only clustering (where multiple CRX instances have identical data, e.g., one authoring instance pushing content to multiple publish instances) requires some load-balancing and caching magic by fine-tuning and properly configuring the Dispatcher tool for the best results.

Active clustering, on the other hand, is used to keep multiple instances in sync. Potentially a lot more labor intensive, this approach might prove to be useful for multiple authoring and publishing instances. Again, load balancing is the key here, as active clustering on its own will merely replicate data across environments without auto-magically improving performance and response times.

"Shared nothing" is one of the latest active clustering variant additions to the CRX, which ensures that all elements of the data store are in sync, without using any shared storage.

There's also an option that Adobe calls "Reverse Replication," in which community-generated content — pushed into the delivery environment (in the DMZ) — is replicated back to a CRX instance inside the firewall, where it may kick off a workflow. Pending any approvals, content then is republished to wikis, blogs, other portions of the sites and/or to different publishing instances. Because of the security restrictions, a server in the DMZ is not allowed to make a connection to a server inside the firewall. So, for reverse replication, instead of the publishing server pushing this content to authoring server, the authoring server regularly polls the publishing server and pulls content from there.

You can set up the system to completely pre-bake everything (the so-called "spidered snapshot" approach) either on a schedule, or one-time-only (to burn an archival DVD, for example). This is a CPU-, bandwidth-, and disk-intensive undertaking, so if you have a lot of static content to burn at frequent refresh intervals, be forewarned.

Adobe has built some impressive cloud-based deployment options in its recent editions. AEM Cloud Manager is a SaaS application for setup and management of cloud deployment options for Amazon EC2-hosted cloud computing environments containing instances of AEM. In the initial release, Amazon was the only supported provider; Rackspace followed several months later.

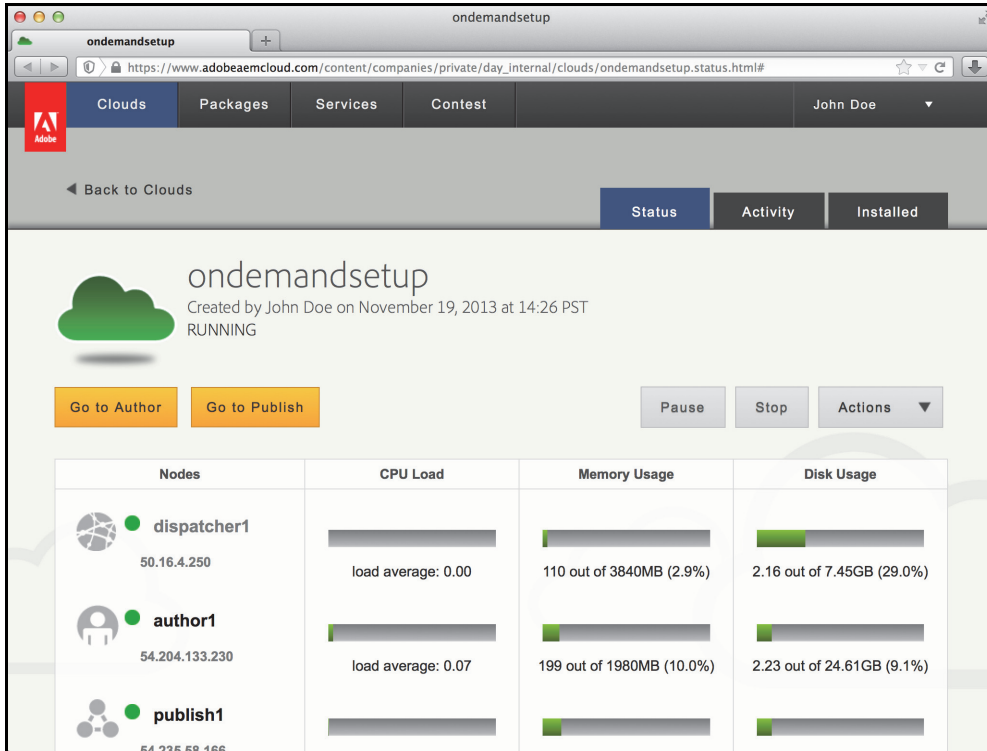


Figure 107. Administration panel in AEM Cloud Manager.

AEM hosted and managed services are offered in a dedicated, single-tenant environment, which is managed either by Adobe or one of its partners. If it's Adobe, your provider choice is limited to Amazon EC2, but you may have more options with a partner. Alternatively, you can go the self-managed route. Note that because Cloud Manager is a separate application, you will not be able to access it from the common AEM UI; you also won't find an LDAP or similar integration for single sign-on. In essence, the more Adobe products you use, the more logins you will have to remember: one for AEM (using an Adobe ID), one for any former Omniture products, and another one for the Cloud Manager.

Adobe's approach to user management is both flexible and potentially complex. Out of the box, AEM's web client provides an uncluttered UI for managing users, groups, and permissions, with most operations requiring only drag and drop, or picking from a drop-down list. The product comes pre-configured with 13 groups with names like "author," "admin," "administrators" (delegated admins), "workflow-editors," and so on. If the presets don't map well to your requirements, you can redefine the groups and permissions or add new ones. Groups can be nested to achieve rights aggregation. For example, if you want all the members of one particular group to have rights on the same objects as another group, you can drag and drop the first group into the second one's membership list. Also note that nodes in the content hierarchy can inherit rights from ancestor nodes. Almost any kind of entitlement system that you want to model can be implemented in AEM through the Security UI — if you're up to the task. Map out your needs thoroughly before you start.

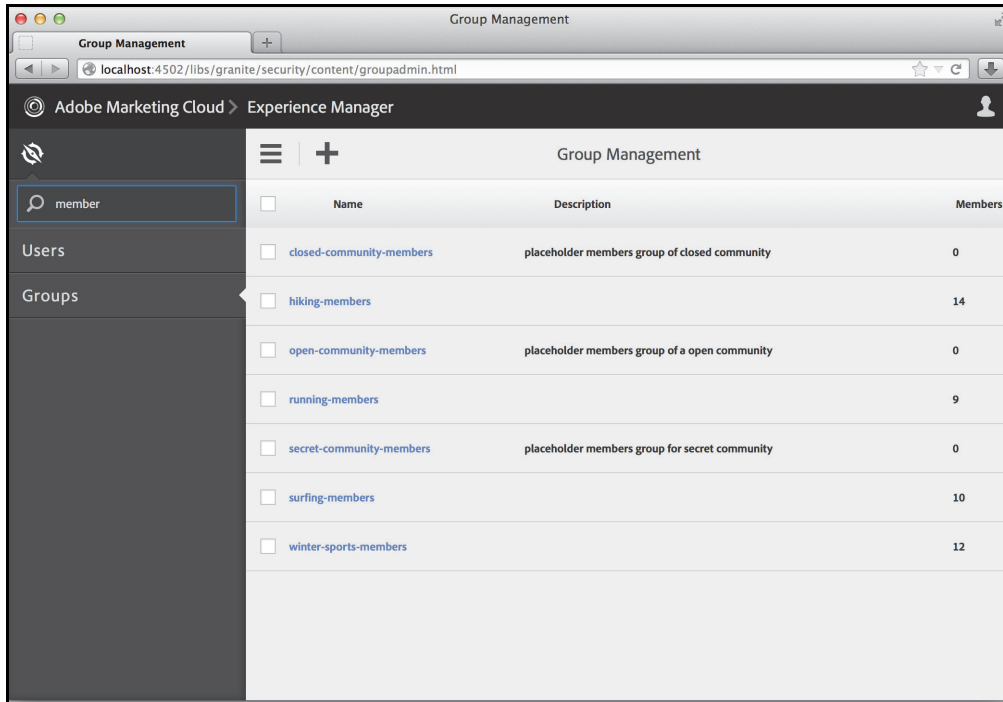


Figure 108. When setting up roles and permissions, it's possible to drag-and-drop groups such that a particular group can contain individuals, other groups, or both.

Notably, the security features include delegated administration as well as impersonation. Impersonation is useful when someone wants to act in the capacity of another person in a workflow (so as not to hold up the flow, in case someone is sick or on vacation), and it's a feature relatively few systems support natively. On the down side, Adobe's implementation of this feature makes it hard to tell who did what. If one account impersonates another, an entry is made in the system audit log when the impersonation starts and ends, but other logs (such as access logs) hold no information about the fact that a given event was done by an impersonator. Therefore, if User B is impersonating User A, all events will look as if they were performed by User A, which is not good from a governance standpoint. We think this is indicative of many AEM features: great concepts with middling execution.

AEM supports pluggable authentication via JAAS (the Java authentication service), meaning that you can swap LDAP for some other type of authentication (e.g., a Kerberos-backed single-sign-on scheme used in your company's portal).

AEM is light on built-in system reporting services, but offers fine-grained logging capability, which makes reporting a potentially tedious exercise in log-parsing. Starting in 5.5, Adobe added JMX for system monitoring. It's not without limitations, but it has obvious advantages. The well-regarded Java Management Extensions (JMX) technology should (in theory) offer more built-in reporting capabilities for the Java VM, but you will want to make sure it is properly set up and configured by an experienced team. You may also want to ready yourself for some development work here, since there are a few Sling components that don't yet contain the JMX hooks. It will be up to you to code that.

Development

Development options in AEM are rich, varied, and complicated. Be wary of the “varied” part. Adobe engineers like to brag about all the different ways to accomplish the same result, but most licensees find this somewhat of a burden. It’s not unusual to see Adobe specialists and partner engineers disagree amongst each other about which approach to take. You need strong program management — and a strong stomach — here.

For the presentation layer, you have two options:

- Traditional Java Server Pages (JSP)
- A new Adobe templating language called “Sightly”

AEM developers will be most familiar with the JSP approach, where they can use most any Java IDE for common development needs. JSPs make use of server-side scripts and taglibs. Customizing or extending these is definitely a job for a developer. That’s a demerit at a time when competitors are making templates easier for non-technical users to create and manage.

Adobe has responded in AEM 6 with Sightly, its very own templating language and framework, which emphasizes layout logic with HTML5 markup, while forcing business logic into external objects. In theory, this makes templates less technical to grok, but your developers may still find Sightly code intimidating — and a steep learning curve. Sightly appears to be well documented, but the tooling around it remains thin, and there’s no large community of peer support behind it, like you would find with JSP.

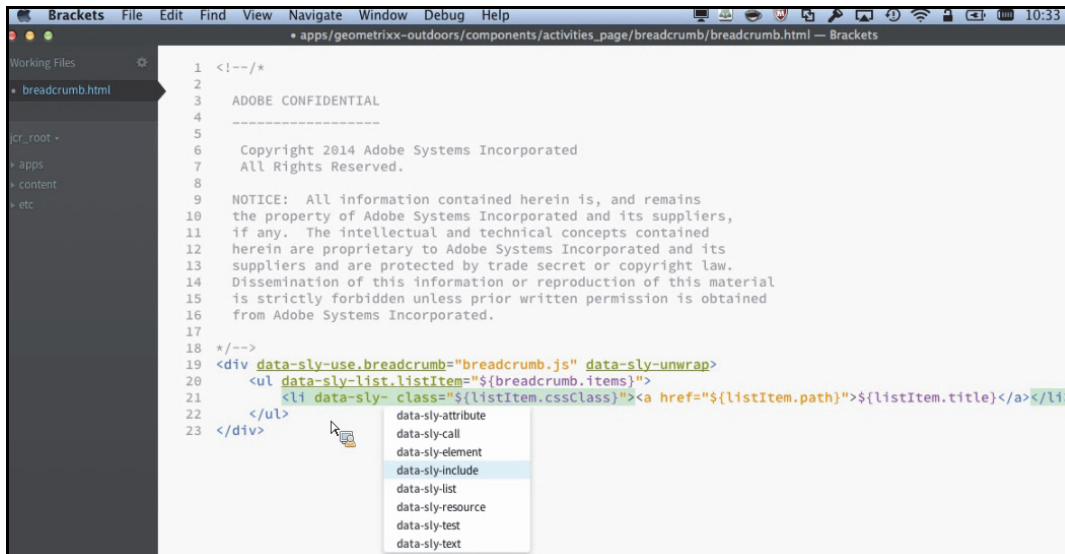


Figure 109. Editing a template using Sightly.

Widget development is more complicated, even despite AEM’s ExtJS-based widget framework. However, Adobe’s developer documentation contains some detailed walkthroughs. The AEM Component architecture is compositional and geared toward reuse of widget-level chunks, allowing the creation of dynamic portlet-like pieces without the pain of full-blown portal development.

Any extensions and customizations done in AEM may present a challenge during the upgrade process; some may require you to run custom upgrade scripts. The majority of customers using

AEM do a fair amount of customizations, since out-of-the-box components clearly cannot cover all scenarios for different organizations. Typically, AEM customers develop custom components extending the foundational components, templates, widgets, overlays, etc.

One customer told us they “basically had to rewrite everything” during one of their upgrades due to the amount of customizations they had developed. While this situation is unlikely to be too common, proper planning and up-to-date development documentation should not be overlooked during an upgrade.

According to another AEM licensee, the theoretically painless swap of existing files with the newest version ones — drop a new .jar into AEM, rename the new .jar, delete the old .jar, restart and you’re on the latest version of AEM — to demonstrate the ease of upgrade may demo and sell well, but this “automagic” doesn’t work that effectively when you’re already working with an existing AEM installation. “Magical tools are there to get you launched, but not to upgrade you,” they concluded.

Implementers and system administrators new to JCR will find the CRX node-and-property system particularly bewildering at first. You’ll need to come to grips with the JCR spec first, and Adobe’s implementation of this spec second. AEM is using the Apache Jackrabbit reference implementation of the JSR 170/283 spec as its foundation, and Jackrabbit is the most complete implementation of the spec. Regardless, you should understand that there are variations in the different ways of implementing the specification, which can add on to the already steep learning curve. Once you’ve gotten a feel for how content is modeled in JSR 170/283 (Day’s documentation offers detailed best practices), repository setup, and administration in CRX begin to make a lot more sense. Still, it likely will be a steep learning curve.

Before authors can browse corporate repositories, a developer must configure each connector, determining how the remote data will map to the web content repository (and of course, setting up user access rights). You’ll want to give part of the process careful thought, because the product wants to display the “virtual” content structure as a kind of site tree the way it does your web content. This may or may not be what you want. Configurations must also be set concerning where and how content is replicated, and whether data is pulled from remote repositories at approval time. In short, all the usual, difficult content integration questions still apply. One customer explained, “Using CRX is not as simple as it looks in the diagram.”

Theoretically, you can use these virtual content views to write data back to remote repositories, but many Adobe customers employ it in read-only mode, depending on whether there are middleware functions that need to have control over writes and deletes (as there often are). Clients do commonly use the virtual-repository feature to apply AEM’s versioning and search capabilities to content that originates from other applications or systems, such as SharePoint. However, if you’re thinking that a few CRX instances, plus some SharePoint connectors will let you manage and consolidate hundreds of SharePoint mini-silos across your enterprise, you’re wrong. This isn’t the solution for your SharePoint proliferation problems. On the other hand, it’s a superior approach over competitors like Oracle that make you import SharePoint libraries before managing and exposing those documents on your website.

Adobe’s connectors — including the one for SharePoint — are not cheap; some of the more popular connectors command US \$30–80,000 each.

Performance

In the management tier, you could traditionally count on 20 to 25 concurrent contributors per CPU when estimating needed server power. Obviously this can vary, depending on types of content and usage intensity. This probably improved a bit in Sites 6.0.

In the delivery tier, AEM is flexible in overall system design, as well as production modes, so that it’s hard to generalize about performance. Caching can occur in several areas: CRX, the management-layer application memory, the Dispatcher Cache, and a reverse-proxy layer. How you decide to configure caching will depend a lot on whether you intend to deliver highly personalized or componentized pages that undergo frequent updates, versus static content. In a worst-case scenario, with highly dynamic, personalized pages assembled from scratch with no caching, Adobe says a single-server system will typically max out at around 30 page-serves per second. Twist all the knobs in the direction of full-on caching, and the system can handle 3000+ hits per second. However, in the latter case, you take a chance of serving stale bits once in a while. Adobe publishes useful guidelines for performance tuning and offers a good deal of advice in its documentation publicly available from <http://dev.day.com/>.

If you intend to add AEM DAM to your system (as many customers do), you’ve introduced even more complications to the performance picture, of course. At the very least, you’ll want to budget extra for multi-core processors and fast RAID arrays for storage of large media files, as well as ingestions and transcoding of video files.

Content

Contributor Experience

Like many of its competitors, Adobe has definitely sided with power users here. It wants to put as much power and features as possible in front of you, with the assumption that you are a dedicated site contributor or marketer. The user experience represents a very poor fit for “casual” contributors, who may get overwhelmed.

AEM Sites now has two user interfaces:

- Default touchscreen
- “Classic” UI

The company clearly put a lot of work into the touchscreen, which far exceeds the typical approval-plus-simple-edits in most vendors’ mobile admin interfaces. With AEM 6, the entire UI has been touch optimized, which is ideal for editors and marketing specialists who work primarily from a tablet or phone.

But what if you work primarily from a desktop interface? The touchscreen UI may represent a step backward for most. There’s no right-clicking and some of the behaviors are unexpected. The classic CQ “sidekick” control panel has been replaced by a set of some ambiguous icons, and it will not run on IE8 (or earlier). Even if you toggle back to the Classic UI, the system can frequently back you up into the touchscreen interface since it’s the default setting. This has

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

generated consternation on Adobe customer boards. Revealingly, the company essentially told its customers to suck it up and work more from mobile devices.

Historically, the “Classic” UI has been met with mixed reviews. Some customers find the experiences comfortable and conducive to getting work done, but others have said various aspects of the UI are a bit clumsy. Tabs are ever present in the interfaces, of course, and in previous versions of the product, dialogs would pop in your face like virtual air bags. The nested-dialog problem is mostly gone now, but recent UI revamps rely heavily on tabs that sometimes are clearly labeled with text and other times just have icons with no obvious meaning until you mouse over them and wait for the tool tip. You’ll want to do your own user testing with this product.

Most of the things you can do in the contributor’s environment start in a small floating palette called the “sidekick,” which encapsulates many commands and offers a powerful way to get to different functionalities quickly.

The sidekick is always hovering in front of the page while you’re working in-context. It looks cool, but in our experience, it tends to get in the way. It was a bit buggy when we used it, and would occasionally get stuck if part of the palette went outside the active window area. It loads in a very leisurely way each time you have to return to the page (e.g., after a preview), which can get annoying very quickly. It is minimizable, but not dockable. Many users actually find it a pain, and you’ll see Adobe engineers constantly moving it out of the way during demos. You’ll be doing that in production, too.

This palette serves multiple purposes. Depending on which tab you pick, you can see:

- A component pick list
- A set of page commands (delete, lock, publish, view properties)
- General information about the page (audit log, permissions)
- Versioning commands
- A tab that contains “Start Workflow,” as well as step-forward and step-back actions if the page is already in a workflow

Note that there is no sidekick in the touchscreen UI.

To be fair, overall usability has improved in AEM over previous versions. The UI certainly feels snappy, and many common operations are supported by drag and drop. Nevertheless, the interfaces generally hide much more information than they expose (they’re rather sparse by industry standards — making them seem deceptively simple in demos), so understanding the system’s true capabilities will take time and training. This is truly a click-heavy, power-user interface with too many windows popping. Budget accordingly for support.

On the plus side, the user interface is easily localizable using a skinning methodology. As you would expect for a Swiss product written in Java, AEM is Unicode compliant at all levels.

Accessibility is decent as well, since the UIs benefit from keyboard shortcuts, ample tool tips, consistent color-coding, and so forth.

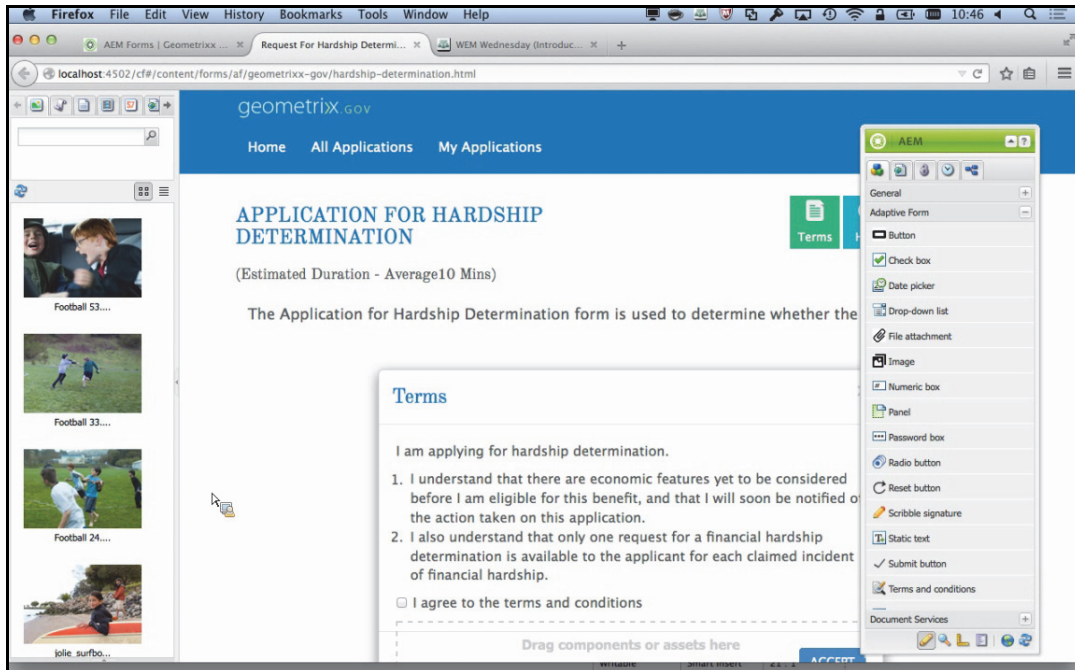


Figure 110. “Sidekick” (right) can take a considerable amount of real estate when in the in-context editing mode.

Contributing Content

For content contributors, there’s a simple WYSIWYG editor, complete with language-appropriate, real-time, spell-check (Figure 111). Note that unlike most other tools, AEM’s rich text editor functionality is not easily configured by role, but rather by template, so anyone editing the same rich text will have access to the same formatting capabilities, even though their roles may differ. Note also that Adobe has elected to build its own rich text editor, rather than employing an industry-standard RTE — as most other systems now do. AEM engineers brag about theirs, but history has not been kind to platforms with homegrown RTEs; you should consider this a long-term risk.

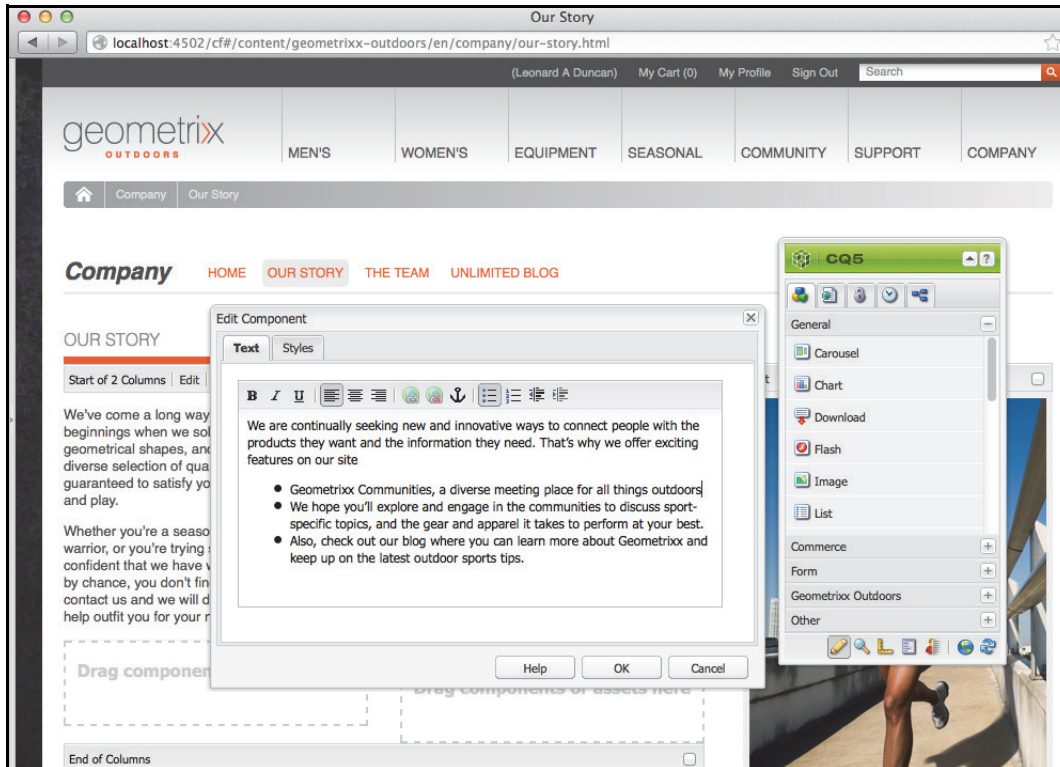


Figure 111. You can do in-place editing or use a rich text editor to edit content. WYSIWYG editing is rather Spartan by default and presents the same editor to all users, regardless of role. A floating sidebar that takes the place of a menu system, exposing or hiding commands through icon-buttons and flip-open dividers is always present during in-context editing.

On the plus side, it works reasonably well (with a few little hiccups) in the touchscreen interface, which is no small feat.

The product also allows in-context editing within the actual CMS, but with serious shortcomings. You cannot browse to a page via a website and login there to work on to begin editing, as you can in most other packages. Also, the in-context editor doesn't indicate permissions in any immediately obvious way — as nearly every other CMS product would. Instead, you need to mouse over a component to see the blue “I can edit” indicator.

Alternatively, content contributors can do wiki-like, in-place content editing by clicking on the placeholder and editing content in the context of the stylesheet. Editing content in this fashion is convenient for occasional users, but it has implications on content reusability and repurposing.

To further collaborative work, you can insert colored sticky-note-like annotations in order to collaborate during the content creation or editing process. Annotations can be enabled only on certain types of components. In addition, there's an ability to use sketches — simple, line graphics to highlight certain portions of the page. While this might be a useful concept at first glance, many users are unwilling to clutter the screen with more moving parts, in addition to the omnipresent “sidekick.

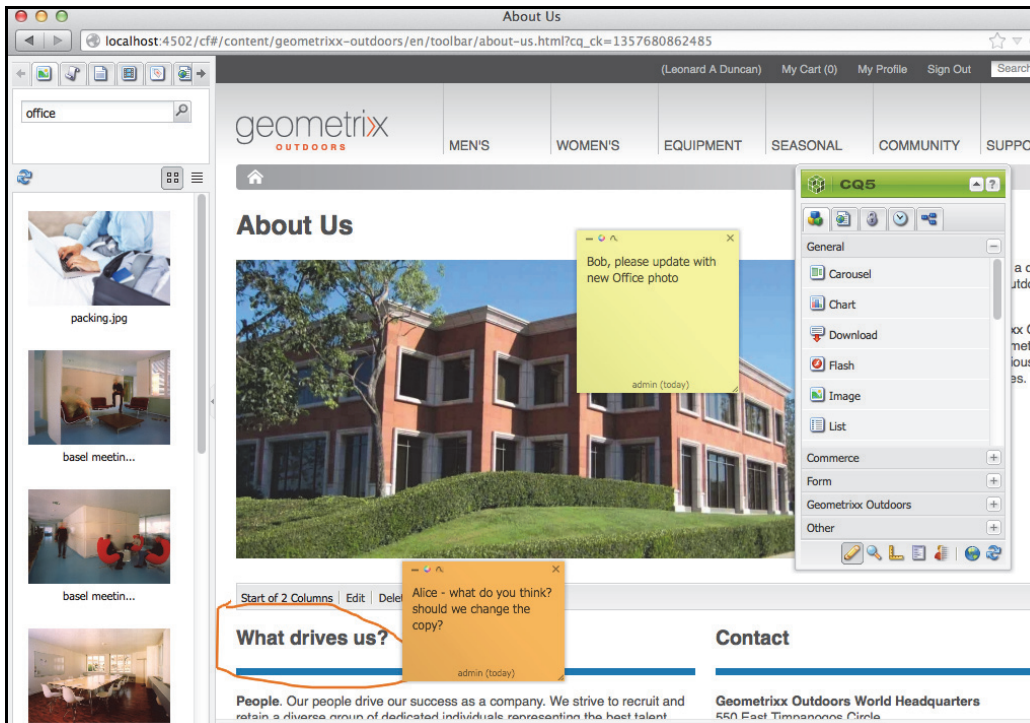


Figure 112. AEM annotations and sketches features are an attempt at making content editing more collaborative and, hopefully, without too much mess on the screen.

AEM Sites 6.0 has extended this with a notion of “projects,” where marketers can organize and track site changes and campaigns better.

With suitable rights, contributors can add custom components to individual pages, a nice feature for companies with a lot of semi-structured content that can be packaged nicely in an AJAX widget. AEM ships with various components out of the box, as well as offering an API for building your own components. Beware though that building custom components most likely will be necessary, which will make your AEM implementation more complicated going forward from the standpoints of upgradeability and maintainability.

Another way to get external content into Adobe is by using the Site Importer functionality, where you can take an existing website and AEM automatically converts it into a project that can be used as the basis for your new application. This is a new functionality, so test it well; the quality of what AEM creates will heavily depend on the existing site and how well it’s structured.

Slotting components into pages requires that you build (or extend existing) templates. Adobe AEM ships a sample site containing a range of templates covering many typical use cases. Starting with these, you can either edit the relevant JSPs or make “coarse-grained” design decisions right on the page, using different layout widgets (e.g., for two- or three-column content areas). When a page is in “Design Mode,” blue header bars appear at the top of each major page element, with an Edit link on each that will take you into a dialog that offers appropriate options for configuring that piece of the page. Depending on how the template was designed (and the user’s rights), you can drag various page elements around, and those that are repositionable will snap into place in drop zones that can accept them. Again, depending on the

implementation, the generated code can contain countless divs and iframe elements, but it's better than the plethora of table elements that similar systems (including earlier versions of AEM) used to generate.

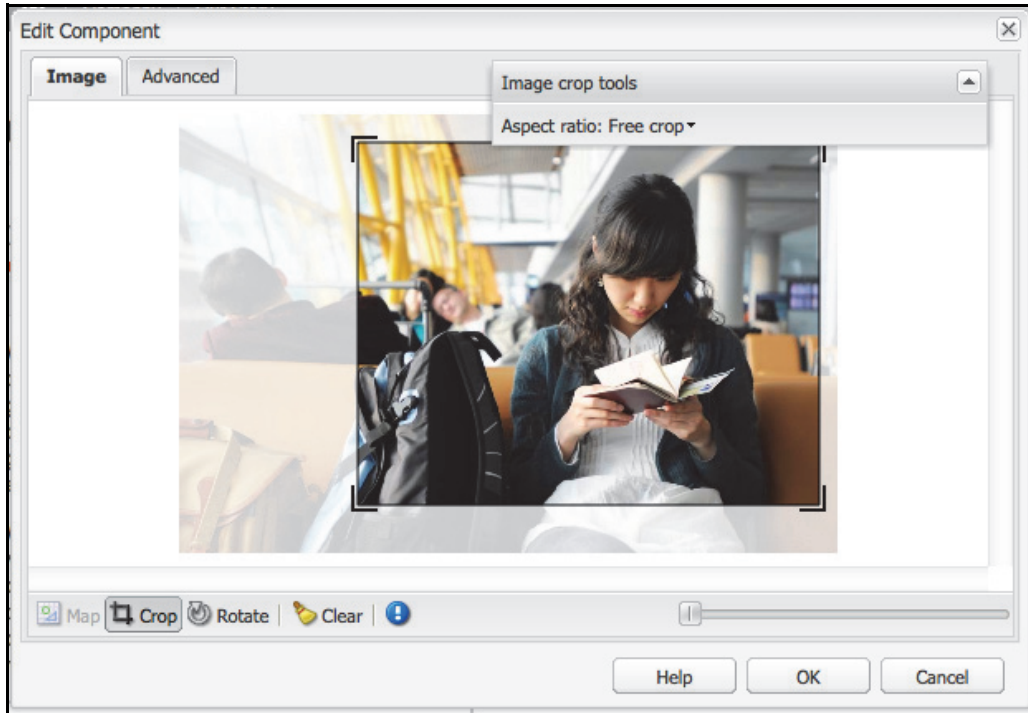


Figure 113. Image editing in a browser, where you can scale, crop images to create a derivative instance of each — without leaving the AEM interface.

Along with WCM, Adobe sells AEM DAM for digital asset management, which goes beyond simple image cropping. AEM DAM is not cheap, but it does integrate well with the company's Web CMS (and there's no need to learn a new client), and the workflow system is the same as for WCM, given that both WCM and DAM work off the same JCR-based content repository. Templates, taxonomies, user interface definitions, workflow models, the "help" subsystem, even system configuration files — all are managed as content items in the repository. This, of course, gives you the ability to apply repository services (versioning, logging and tracking, and so on) to these facilities. It also means that the derivative entity natively becomes a new asset in the repository, itself. You should note that when a designer modifies the source file, those changes cascade throughout all of the derivative versions of that image employed across your various web properties. You can version your Sling scripts this way, for example, or put new workflow definition through an approval workflow. Again, you'll want to model this carefully, lest you build an administration nightmare.

As for metadata, a non-technical person with suitable rights can add new facets and otherwise manage a hierarchical taxonomy without great difficulty. However (as with Oracle), you are limited to just one tree, which is inconvenient for enterprises with multifaceted taxonomies.

Note that rendering is done on demand, at run time, so if you'll be using these sorts of capabilities a lot, you'll want to consider the CPU implications and budget accordingly.

Content Lifecycle

The product’s workflow capabilities are generally at about the same level as others at this market tier, with it being somewhat easier and more flexible to design and administer, and customize the workflows. However, because this is AEM, the workflow is quirky.

AEM’s graphical workflow designer allows business users to create process flows using drag-and-drop gestures. Activities or tasks (called “steps”) can be one of three types: participant, process, or container. A participant step is one involving human interaction. A process step hands off control flow to a custom script or Java class of your creation. This means that, if necessary, you can bend the system to BPM-style uses. A *container* step calls another workflow process, and therefore supports potential reuse of workflow logic.

The Sites workflow system will accommodate serial approval flows as well as parallel flows with joins. When a flow splits into two parallel branches, the split can be of an AND nature or an OR nature. The former means *both* paths will be followed; the latter means only one control-flow path will be followed, based on a conditional expression that someone with a bit of JavaScript knowledge can insert in a text field in the OR split’s property list.

There are limits to what the system will do, of course. While the system natively supports time-outs, it does not support retries. Nested workflows were changed in 5.4 to support arbitrary levels of nesting, with splits into arbitrary number of branching — all in the attempt to provide more flexibility for customers looking for more sophisticated workflows than a two-step approval process.

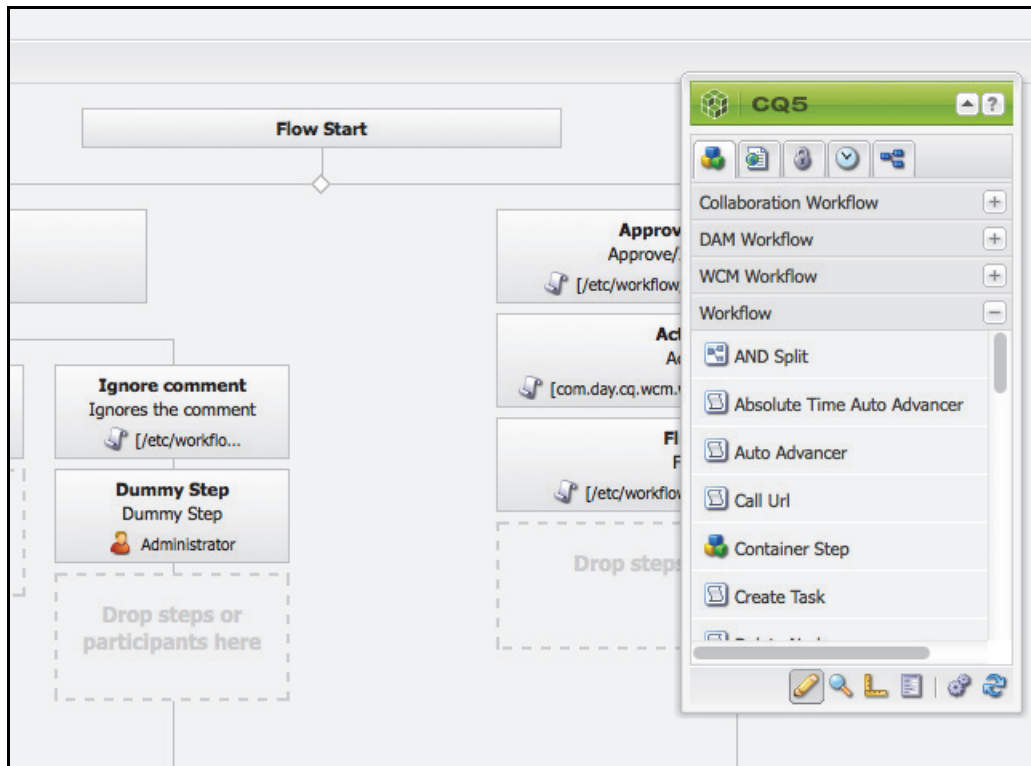


Figure 115. Modeling workflow in the AEM Workflow Editor interface.

Adobe added an iPad/iPhone application (AEM GO) for on-the-go workflow participation. It was built as both a native and a hybrid app, with the expected touch user interface and email client-like inbox management features. These services are newer and a bit finicky in practice, so test carefully.

As you might expect, AEM has decent globalization features. Authors can trigger international workflows and pour localized text and images into different skins that form the specific templates a company may use in different national web properties. The “Multi Site Manager” tool, targeted at large multinational corporations, allows licensees to treat different levels within multiple site hierarchies as individual objects with inheritance. Following SDL’s BluePrinting approach — but not as visually appealing — this methodology allows content managers and designers to apply different rules, layouts, and features at different levels across multiple sites, and have changes optionally cascade down through hierarchies.

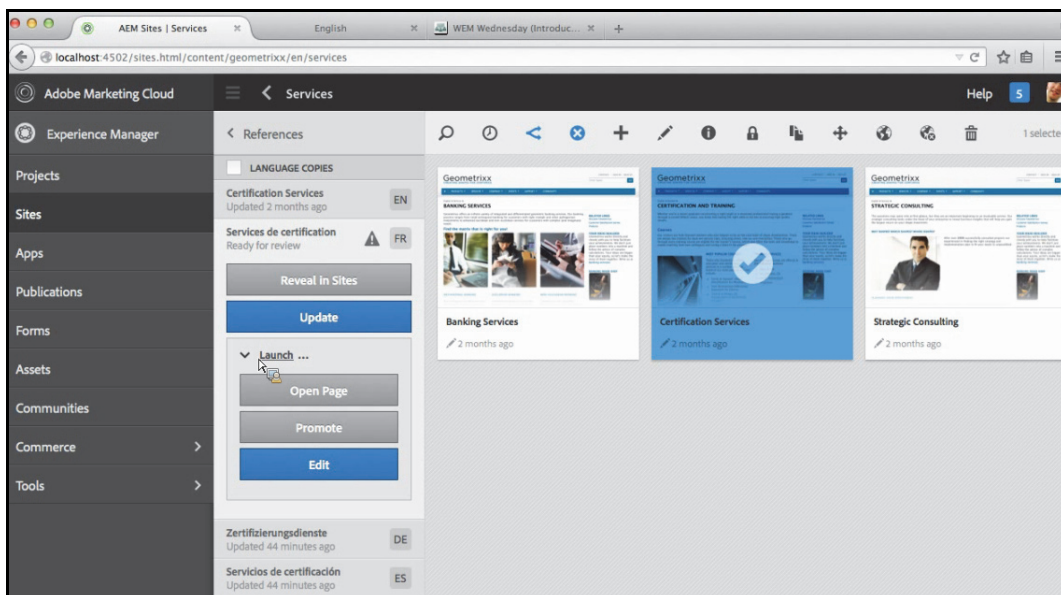


Figure 116. Adobe’s content localization interface is more business friendly than before. Note that the new interface relies considerably on iconography and now has a relatively lower-contrast interface.

AEM Sites 6.0 improved some of the business usability of multisite management — particularly by exposing dependencies among sites and content items better. However, you still probably won’t find it quite as slick as what you’d find in SDL Tridion.

On the plus side, AEM’s broader “packaging” feature is quite powerful. Authors, designers, and developers can easily assemble ad hoc collections of content, images, templates, and even workflows and code into a single package, then export the package as a self-contained unit and redeploy it into another instance of AEM. This can be important for rolling out new sections of sites or upgrading a collection of custom JSPs, or reconciling configurations across sites. It’s also nicely tied into Sites’ new “Projects” feature.

For retention purposes, the product makes snapshot-style archiving of an AEM installation quite painless. Creating a comprehensive snapshot of the entire system, including not just repository content but configurations, logs, executables, license files, and system artifacts of every sort, is a push-button affair — all are archived into a single file that can later be double-

clicked to explode the entire archive into an exact copy of the working site, exactly as it was, including all deployed components. In other words, you've made your site into an installer that can be run anywhere to recreate your site and the software that runs it.

Experience

Publishing

Search-engine friendly and “vanity” URLs are the norm in AEM, and the contributor UI exposes a simple interface for specifying page titles that later become part of the URL path. Unlike simpler tools, there's no native vanity redirect features (although you can get a developer to create one for you).

The Mobile add-on, is used to render content in layouts appropriate to various smartphones and tablets, using emulators for preview of content display on specific device types. The emulators are comparatively quite spiffy, and you can feed them different parameters to change the view. As always, however, no mobile emulator will give you 100% fidelity, particularly for Android devices.

AEM has minimal native services to publish to social media channels like Facebook and Twitter, though you can build your own and doubtless, many have. It is an example of the more trivial (but useful) features that go missing in a more toolkit-like platform such as AEM.

Experience	
Publishing	
Standards Adherence	
Multichannel	
Mobile	
Digital Marketing	
Site & Campaign Analytics	
Testing & Optimization	
Segmentation & Personalization	
Social Media Integration	
Promotional Campaigns	
Community & UCG	
Workplace	
Collaboration & Networking	
Dashboard	
Ancillary	
Site Search	
Online Forms	
Module Ecosystem	

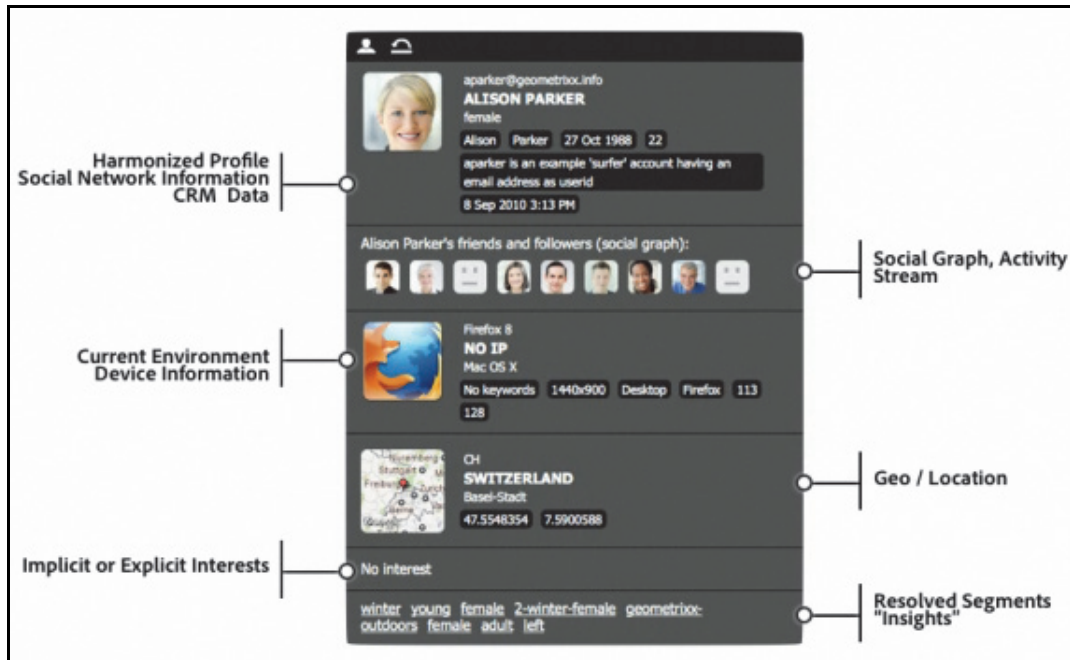


Figure 117. A breakdown of the data that goes into “Client Context” in AEM.

Digital Marketing

Adobe offers some personalization services via a separately licensed module called AEM Targeting. AEM Targeting has features for segmentation, campaigns and targeting, which caters to implicit or explicit personalization use cases. It essentially allows you to create segments of users based on their activity, profile data, and other parameters. You create a segment using a browser-based segment editor.

The segment editor appears to be targeted for casual users but not power users. Creating a segment based on complex criteria could be very cumbersome using a browser-based, drag-and-drop interface; test carefully (Oracle does this much better). Once you create a segment, you create Campaigns and Teasers that eventually allow you to display relevant content to different segments of people.

A closely associated concept in AEM is that of “Client Context,” which provides you with information about a page as well as the current visitor. This information includes tags on that page, the number of times the user clicked a specific tag, and the user’s technical data — e.g., browser, OS, the segments that resolve for the logged in user, and so forth. This information is stored as a set of key value pairs. You can programmatically retrieve this data using a JavaScript object and based on that, you can personalize the site.

AEM provides its own multivariate site testing to help you decide which component is more suitable or relevant for your audience. When visitors browse a page, a variant of a specific component (an image or a banner) displays. For each component, AEM calculates the number of impressions shown and the number of click-throughs on those impressions. Based on these, a Click Through Rate (CTR) is calculated, which is essentially the ratio of impressions to click-throughs. A higher CTR means that visitors click on that component more often. When showing different variants of a component, you can increase or decrease the frequency of a

component by changing its weight, also called a “bias” in AEM lingo. Not too advanced, but it works.

Likely because Adobe bought Omniture right before the Day acquisition, AEM’s targeting and analytics capabilities are expanding with incremental integrations between AEM and Omniture’s SiteCatalyst and Test&Target.

In the 5.5 release, Adobe came up with the new Analytics Framework based on an integration with SiteCatalyst, and it improved its existing minimal integration with Test&Target. One of the enhancements that customers find valuable was addressing the pain point of having to tag the entire website to map values to SiteCatalyst. Now, tagging can be done in the UI, and you can create mappings between your site and the desired reports directly. On the CMS side, that means JavaScript is generated and must be run for the tagged components. This is still a task that is far from trivial and is better handled by a seasoned analytics professional; it is unlikely to be enjoyed by your marketing department.

The only reason to use the Test&Target integration is when segmentation and personalization rules provided in AEM Targeting are unsatisfactory and you require a much (much!) more complex setup. Because these tools are not integrated, you’ll have to work in separate interfaces in this case, but AEM will still be able to digest the rules you set up in Test&Target.

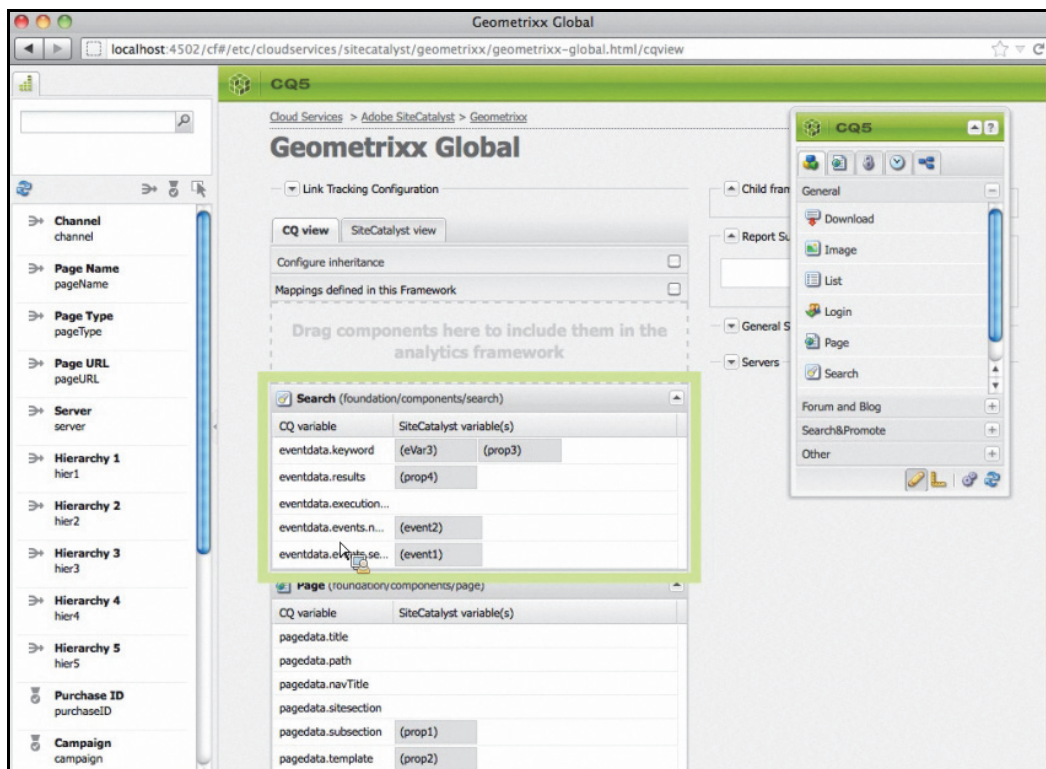


Figure 118. Link Tracking Configuration interface for SiteCatalyst in AEM.

Note that SiteCatalyst and Test&Target are not inexpensive solutions. In our experience, customers of both offerings tend to exploit only a fraction of their capabilities. Be wary of Adobe demos that assume you already have licenses to them — particularly if you do not — as both are quite labor-intensive offerings.

Adobe sells an optional AEM module called Marketing Campaign Management (MCM) that was developed in response to customers wanting to see more features that are not just techy and geeky, but are actually sought after and used by marketers. The idea behind MCM as a marketing automation tool is to allow marketers to plan and manage multichannel campaigns, leads, and deliver email marketing personalized newsletters.

In version 5.6, the vendor made some product enhancements on the MCM side. One of the most notable ones is the ability to create one-off landing pages in AEM using a new Design Importer tool. What the importer tool allows you to do is import a .zip file containing HTML, CSS, and related resources into the WCM and map these elements to editable AEM components on pages. In essence, you will do some form of templating, but without rolling up your Java and Maven sleeves.

Adobe calls this approach a Blank Canvas template. There are a few peculiarities here, mainly that you cannot save this new blank canvas as a new real template. Adobe considers these as “throw-away pages,” and their main purpose is to allow you to change components, but not the templates and layouts themselves. Overall, this can be a complicated endeavor and a temporary solution — especially since you have to create a completely new canvas from scratch if you need to start over.

With this addition, Adobe hopes to simplify page management and expects that any content author and marketing manager will be able to become a web designer (or avoid dealing with IT), by importing bits of HTML and CSS directly into any page and hit the road running with uniquely designed landing pages for various marketing campaigns. (You should note that the Design Importer tool thus far is only working for landing pages — not across the entire AEM instance.)

This idea may appear sound good on paper — and some customers are even under the impression that Adobe is creating a simpler version of AEM. However, WCM history teaches us that importing arbitrary pieces of code into pages typically doesn’t end well. While Adobe’s tools are more sophisticated than those in the beginning of the industry in the early 1990s, carefully evaluate this proposition from a risk and code-cleanliness perspective.

Thus far, we generally have not seen great adoption of this MCM toolset among AEM customers.

Ancillary Services

For both repository and site search, Adobe uses a highly customized version of the well-regarded Apache Lucene, integrated with the Sling and Jackrabbit infrastructure. It supports Google query syntax out of the box. XPath and SQL are available natively, as well.

The “Social Collaboration” module supports blogs with commenting, RSS feed generation, shared calendars, user profiles, moderation, discussion forums, and full-text search, plus wiki functionality that includes management of users and groups in the CRX repository. Wiki content is versioned and searchable, and can be managed like any other content.

Adobe differs from Drupal and Sitecore in one key area: there’s no real module or add-on ecosystem. To be sure, Adobe integration partners have built innovating hosting and support options, but you don’t see any third-party plugins, and we sense that Adobe’s DNA might tend to work against a partner model for this; Adobe likes to do things its own way. For larger customers who may not trust third-party code, this may be such a big deal, but it does tie your horse more closely to a single vendor than some other platforms do.

Vendor Intangibles

Historically, Day had a habit of emphasizing technical vision over execution at the operational, sales, and service levels. For the most part, we see that pattern continuing today under Adobe.

Let's turn to support, first. Adobe's support and bug tracking system have been described by some as rigid, although customers speak fondly of the ability to get through to real product developers. The company maintains an incident-tracking system built on its own CRX product. Product documentation is freely available on the company's website, but not all of it is as detailed as you might desire. However, Adobe remains one of a very few vendors in the WCM space that provides such easy access to its documentation.

AEM customers never fail to mention that this product is not designed for the faint of heart. The learning curve for in-house developers bears a substantial investment of time and money, which is an important consideration since it remains a challenge to find experienced AEM developers. Adobe has recently started pumping up its own professional services here, but they may be climbing the same steep curves. In any case, you'll want to plan to train your own Java specialists in Adobe's technology.

Organizations implementing AEM also frequently turn to system integrators (SIs) who have AEM experience and are supported by Adobe as "official" partners. While some SIs do a decent job, they are not much cheaper than engaging Adobe's own Professional Services, nor are they all created equal. If you're going down the SI path, do your homework diligently, and triple-check the SI's credentials and references to avoid having to switch SIs mid-project. You want to avoid running into the additional expense of hiring Adobe's own consultants to supervise your implementation and act in a subject-matter expert capacity; the reality is that most major implementations seem to turn out this way.

Choosing the right integration partner in combination with relying on Adobe's professional services seems to present the safest option, but enterprises that have taken this route report that the two teams will often differ on fundamental issues of architecture, development approach, and even terminology. In AEM, Adobe may have built an impressive cathedral, but their clergy does not preach from the same Bible. In any case, assume that initial implementation costs will run 4-8 times your licensing fees.

For licensing, AEM pricing is per installed server instance, rather than per CPU or per user as it is with other vendors. AEM starts at US \$120,000 for a basic two-server setup comprised of one "authoring" server (behind the firewall), and one production (DMZ) server. We don't know of any recent customers who have licensed that small an AEM estate, but it gives you a sense for where you're headed; note our cautions below about negotiations.

There are positive things about Adobe's approach. Within a given server, you're free to use as many CPUs as you want. You don't have to pay anything for using the system on staging, testing, or development servers. However, add-on modules do not come cheaply: AEM Social Collaboration, Targeting, and Marketing Campaigns Manager run \$50,000 to \$90,000 additional (each). AEM Assets as a standalone product costs \$90,000 (for the base package, which includes licenses for one server, five named users, and a limit of 50,000 stored assets before extra charges apply).

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

Some customers and prospects are often confused about Adobe licensing, and dependencies with DAM and WCM in particular, sometimes falling under the impression that DAM is a mandatory component to WCM. This is not true; carefully review your quotes from Adobe and insist on licensing only the components you actually need for your organization.

For licensing the cloud version, you're looking at an annual subscription-based invoice, where pricing is based on metrics you estimate in advance, thus committing to certain year-long capacity based on instances you think you might need during that year. The "use or lose it" rule applies here.

Plan on budgeting \$25,000 for training (most of that for developer training), and a hefty 25 percent of total license fees for first year of maintenance and support. (That's for 8/5 support. Figure an additional five percent for 24/7 support.) We'll say it again: Adobe brings a comparatively high total cost of ownership.

Adobe seems to prefer a smaller collection of well-heeled customers over numerous speculative deals involving smaller customers, so don't be surprised if their sales team seems less aggressive than some of its competitors (unless your purse strings are long). However, like others in the WCM space, Adobe sometimes makes exceptions and foregoes margins to sell the product in favorable customer deals. Regardless, prepare for highly variable experiences dealing with different parts of the Adobe empire.

Conclusion

As a platform, AEM boasts a strong technical vision and a dedication to industry standards around content management. Few Web CMS solutions can claim to offer AEM's combination of extensibility, standards compliance, globalization, experience management potential, and enterprise readiness (in terms of connectivity, scalability, security, and other common IT requirements) — built on an elegant architecture that benefits from huge helpings of open source code. Finally, it has an increasingly deep and broad cloud story.

However, you also have a product that's comparatively quite expensive and frequently difficult to use, sold by a highly disorganized vendor with a very thin track record of supporting enterprise server software, accompanied uneasily by a consulting partner channel for which the best thing we can say is that they tend to be highly stretched and persistently under experienced.

Therefore, if you decide to pursue AEM, be sure to put a strong program management team in place and make sure your contracts are written in such a way that you retain ownership of custom-developed code; you'll probably end up having a lot of it. Remember, licensing costs are only the beginning of your AEM outlay, and implementation costs will represent by far your larger investment. Additionally, don't underestimate your hardware needs (especially if you plan to use AEM DAM). Run the TCO numbers carefully.

To be sure, cost may not present an issue for some AEM licensees. "We go to large customers with public-facing, brand-sensitive sites," explained one Adobe sales person. Large customers value architectural flexibility, but it's hard to mesh Adobe's strategy with its preference for investing in Apache infrastructure projects in lieu of business-oriented content management functionality. Your digital marketing manager may like this product much less than your developers. Baked deeply into the culture of the AEM team is a priority on improving the lot of CMS engineers — not CMS users.

Like many of its competitors, typical AEM customers struggle to gain full value from their WCM investment. The level and sophistication of the resources that are required to achieve success with this platform should never be underestimated. However, with enough time, money, and expertise, you can make almost anything possible with AEM.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

CoreMedia: CMS

coremedia.com

Vendor at a Glance

Specsheet	CoreMedia CMS 6.0 Summary
Geography	Mostly Europe (particularly Germany), some US, Asia
What's New	<ul style="list-style-type: none"> • HTML5 editorial interface "CoreMedia Studio" mostly replaces thick Java client • Introduction of a CDN architecture in effort to reduce content delivery overhead • Early stages of Adaptive Personalization component for creation and management of personalization rules
Strengths	<ul style="list-style-type: none"> • Mature, scalable product with many original developers still in place • Flexible Java architecture • Portal-friendly (integrates with many popular portal products) • Strong Media industry focus, with experience with large deployments • Streamlined in-context editing user interface • Strong globalization features • Company has a comparatively more deliberate product release cycle
Weaknesses	<ul style="list-style-type: none"> • Potentially hardware-intensive compared to other systems • Combination of CPU and seat-based licensing makes the product comparatively more expensive on larger implementations • Doesn't adhere to Java standards as much as some competitors; internal communication relies on dated CORBA protocols • Some important integrations (e.g., MS Office, Lotus Notes) have not been productized and remain third-party customizations • Some important functions are split between two separate interfaces, potentially confusing users • Product is poorly suited for Digital Workplace scenarios • Comparatively lighter footprint outside of Germany
Potential Fit	Global Enterprise, Multichannel Publishing, Ultra-Large Single, Mobile
Unlikely Fit	Informational Site, Basic Digital Marketing, Community Oriented
Compare To	Adobe, Oracle, SDL, OpenText, FirstSpirit, Magnolia, CCI
O S	Windows/Vista, Linux, Solaris, IBM AIX, HP-UX
Repository	Database: Oracle, DB/2, Microsoft SQL Server, PostgreSQL, MySQL5
Client	IE and Firefox are the only supported browsers
App Platform	Java: Can be deployed to a major J2EE app server or servlet container
Licensing	CPU based at \$28,000 per core, plus per-user fee on a sliding scale
Ownership	Privately held, VC financed — largely held by founders

Summary

This is one of these Web CMS products that started out explicitly targeting the media industry. Today, the company goes to great pains to point out that it has large, well-known accounts in other sectors as well (which it does), but media still remains a stronghold.

Over a decade old, CoreMedia’s offering is one of few Java products in this report that pre-dates J2EE. Through a long, gradual, organic process, the CoreMedia CMS has grown into a somewhat intricate — albeit architecturally clean — product that adheres to some well-known Java standards, such as JMX and JavaServer Faces. Conversely, it also sticks with some quite dated standards, like CORBA. The vendor vehemently disagrees with our assessment of CORBA as a dated protocol, but real-life experience shows that in today’s market, you would be hard pressed to find developers and architects who are well-versed in this ancient technology.

Compared to other Java-based vendors, CoreMedia CMS boasts some rather sophisticated user interfaces. The company has invested heavily in making its system easier for different types of authors. As always, you should test any system with actual contributors from your enterprise.

Scalability and overall product maturity are strong points. Consider CoreMedia for complex, media-oriented scenarios or Multichannel Publishing in multiple languages across many sites — including mobile. Take careful note of the per-seat pricing, which comes on top of per-CPU pricing, and often drives licensing costs high for large projects.

For simpler scenarios, CoreMedia is overkill. Also, it’s not a good fit for Digital Workplace Scenarios. The CoreMedia staff seems strongest when working on public sites, rather than intranets.

Scenario Fits	
Simpler Site	
Informational	<input type="radio"/>
Microsites & Landing	<input checked="" type="radio"/>
Mid-Range	
Basic Digital Marketing	<input type="radio"/>
Mobile Site	<input checked="" type="radio"/>
Community Oriented	<input type="radio"/>
Complex Site	
Advanced Marketing Portal	<input checked="" type="radio"/>
Global Enterprise	<input checked="" type="radio"/>
Multichannel Publishing	<input checked="" type="radio"/>
Ultra-Large Single	<input checked="" type="radio"/>
Digital Workplace	
SMB/Departmental	<input type="radio"/>
Global/Enterprise	<input type="radio"/>

Introduction

CoreMedia dates back to 1996, when it was founded under the name “HigherOrder” by a group of professors and students at Hamburg University in Germany. The company has since grown at a slow but steady pace; in North America, there’s a San Francisco office currently staffed by 21 people. The company is making a serious bid for the North American market; as we have seen with other vendors however, crossing the Atlantic can be difficult.

The company has about 170 employees and can boast scores of (often quite large) customers, mostly based in Europe (including Nokia, but also German television stations ARD and ZDF). If nothing else, the company can claim stability as a differentiator; some of the product’s original developers remain working at the company, and at least one of CoreMedia’s accounts has been with the firm since 1996. The most important change in the company’s ranks has

been appointing a new CEO in July 2009, Gerrit Kolb, who came from Sybase, where he was responsible for the North American channel.

Like other successful European vendors, CoreMedia has focused on building a network of partners. The company has invested heavily in creating a partner solution catalog, which is an exchange platform for modules built by partners. These can be modest add-on features or comprehensive solutions (e.g., web-based management of advertising material). Many of these solutions have gone through CoreMedia internal QA testing and code review.

In January 2011, CoreMedia released version 6.0, which brought a new user interface — CoreMedia Studio. Compared to the long-existing thick Java client that CoreMedia nurtured throughout most of the company’s existence, moving entirely to an HTML5, web-based interface was a big step, which introduced considerable usability improvements to the previous UI that was starting to look very dated and kludgy. Some remnants of the thick Java client still remain in Site Manager, which now has limited applicability only for user and site administration. According to the vendor, version 7.0 should see the light of day in early 2013.

Technology

Technical Administration & Security

CoreMedia is a Java-based CMS, only partly adhering to contemporary standards. All servers, clients, and system applications within this CMS communicate using the somewhat archaic, heavyweight CORBA protocol (which pre-dates Java). CoreMedia argues vehemently that CORBA is the right protocol for this particular job; however, customers have raised issues around supportability and expertise with respect to this antiquated approach. Many CoreMedia components, on the other hand, are JMX (Java Management Extensions) agents — JMX being a relatively young, yet well-regarded standard for facilitating the hot deployment, monitoring, and management of runtime objects.

Most customers use Oracle as the repository, running on either Solaris or Linux. Other platforms are supported, such as Windows with SQL Server. You can also choose the open source options PostgreSQL and SQL5.

Note that CoreMedia’s engineering is generally held in high esteem. One customer has said that the CMS had the fewest defects of any enterprise system they had ever encountered, finding only one or two minor issues after intensive testing.

CoreMedia is a “frying” system. Architecturally, it’s divided into two parts: the Content Management Environment for content creation, management, and testing; and the Content Delivery Environment for content delivery to the site visitor (Figure 119). A Publisher component deploys content from the management side to the delivery side. Additional components (such as the Search Engine) operate in both environments.

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

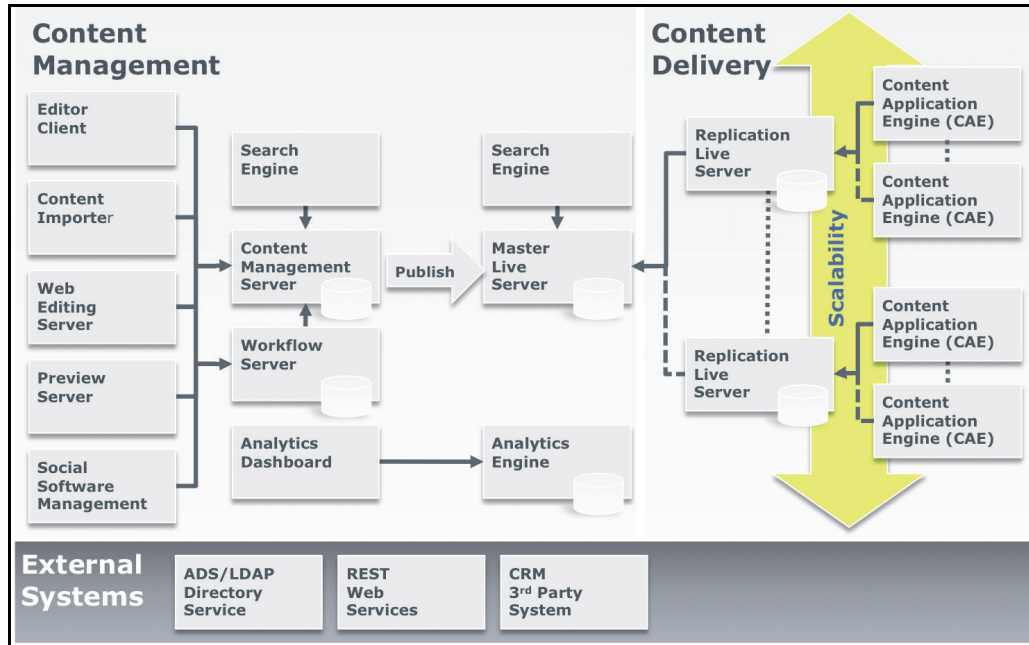


Figure 119. The CoreMedia architecture is divided into two parts: a Content Management tier (left) behind the firewall, a Content Delivery tier ahead of the firewall but potentially fronted by a load balancer, reverse proxy, and/or additional firewalls. (Source: CoreMedia.)

The product has browser-based interfaces for content deployment from the management to the delivery environment. Deployment can be done on a scheduled basis, incrementally as a workflow step, or manually.

Also on the content delivery side, you will probably be hearing more noise about cloud and hosted/SaaS flavors of CoreMedia with the company’s recent cooperative effort with IBS (Internet Broadcasting Systems). IBS is one of its partners and is a provider of online media solutions for US TV stations. IBS is developing and supporting hosted CoreMedia environments via a product called ibPublish for its media and publishing clients. We advise that you carefully research the nuts and bolts of this offering and understand all limitations before signing any deals.

The system provides impressively deep role management capabilities through integration with LDAP directories (including Active Directory). Each object type can have different rights for users or groups. A group can belong to another group, thereby enabling nested and inherited rights. Unfortunately, the well-intended security features make it difficult to figure out who has access to what and consequently, the usability for managing the roles is not great.

Development

CoreMedia CMS is not a simple system, which is why the company has offered a “StarterKit” since the 2008 release. This is a collection of sample content types, templates, and functionality, aimed at making it easier to start new projects using the CoreMedia CMS. It is, in essence, a reference-implementation site, with everything from basic navigation in three languages to live, pre-configured analytics.

You build templates using JSP with taglibs, or ASP.NET. Although the system seems to favor Java, CoreMedia claims to have two customers using an exotic .NET-Java combination. If you'll be exploiting the system's JSP development model extensively, note that the Model-View-Controller paradigm chosen by CoreMedia is JSF (JavaServer Faces). JSF development can be challenging, but CoreMedia makes the process a good bit easier by providing an Editing Services API layer that has pre-built UI components and services. If you were familiar with CoreMedia before version 6, you're in for a new learning experience. Starting with version 6, the APIs went through major changes and you will most likely require a bit of training before delving into it.

For template developers, CoreMedia supports the IDE of your choice, e.g., IntelliJ IDEA, or Eclipse, all of which can use tag auto-completion based on the CoreMedia taglib. Essentially, templates are created by a developer or template designer in an offline development environment. The files then reside and are deployed on the file system together with any other system components. Alternatively — and in a more controlled fashion — you can use Apache Maven for development and build distribution by plugging in the vendor's repository and combining the builds with your own code. Still, some customers lament how much configuration work is needed when working with Maven (and you must use Maven if you want to use CoreMedia efficiently).

SAP customers may appreciate the fact that CoreMedia is one of few CMS vendors that can claim tight integration with the SAP NetWeaver Portal. Again, this reflects CoreMedia's strong presence in Germany, since SAP is a German company. Another German CMS vendor e-Spirit also offers SAP integration. If this is a requirement for you, you'd do well to compare the two approaches.

e-Spirit uses its FirstSpirit CMS interface to edit and manage content; it is then published to the portal, and stored and published through NetWeaver. The advantage is that the CMS content is tightly integrated with the portal (for instance, it will immediately be searchable). By contrast, CoreMedia employs iViews (SAP's portlet equivalent to integrate content editing and presentation fully on the UI surface. However, CoreMedia completely handles the editing and presentation of its content. The advantage is that in this approach, you can add any kind of content CoreMedia can handle to your SAP portal — not just text, but also images and video. The drawback is that the content isn't integrated into the portal. For example, you'll have to jump some hoops to make the CoreMedia content searchable for NetWeaver users. In addition, the vendor has experience with portal integration to IBM WebSphere Portal Server.

Performance

CoreMedia offers caching of application logic, controlled on both a time and event basis, and the cache can be flushed on an area basis, rather than just page- and site-wide. Some of the company's biggest customers are rather demanding media firms with high-performance sites; consequently, CoreMedia can legitimately claim experience in this area. One customer mentioned that the company's background in these large deployments (and integrating with Akamai's global distributed delivery network) might be one of its main assets.

In version 6, CoreMedia continued to meet customers' demands for reduced delivery infrastructure overhead, scalability, and performance by devising CDN (Content Delivery Network) architectures. The CDN Cache Connector allows you to insert a CDN layer (Edge Server, or Akamai-like delivery) between CoreMedia's load-balanced Content Delivery environment and your website visitors. The goal here is to reduce time-to-web, while taking

advantage of globally distributed “cloud-enabled” caches. This approach does appear to be one of the most well-reasoned architectural improvements to the system. However, test carefully to ensure that it will work in your setup and configuration.

In the delivery environment, you are required to utilize CoreMedia’s “Replication Live Server,” which can be put on as many nodes as required. It is essentially the delivery portion of the product without the management overhead. Each of these “Replication Live Servers” runs one or more Content Application Engines, which deliver content. So, a CoreMedia rollout can be hardware intensive. Meanwhile, replication layers act autonomously and allow for automatic re-sync in the event of a temporary disconnect, which can be an important safety net in a horizontally scaled and distributed content delivery infrastructure.

Content

Contributor Experience

Over the past years, CoreMedia has invested heavily in usability. Nevertheless, as with Adobe Experience Manager, occasional users might find some interfaces (other than the in-site editing) to be intimidating. This is something you’ll want to test.

With respect to internationalization, all user interfaces ship in English, German, and French. CoreMedia uses the standard Java mechanism of language-specific resource bundles that you can modify as necessary. Customers can translate the resource bundles themselves, but cannot sign the bundles, which is required for using Java Web Start. This means that users may get a security warning when they log into the interface. As an optional extra, licensees can buy translations to Chinese, Spanish, Italian, Russian, Portuguese, Japanese, Thai, Arabic, Farsi, and Polish interfaces.

As of version 6, the main editorial interface transitioned from Java-based thick client to a web-based HTML5 environment dubbed CoreMedia Studio, following industry trends for less reliance on IT, mobility, and cloud, with accessibility from anywhere, any time.

While most customers may appreciate the HTML5 interface, you should carefully check what has been lost in this transition. In particular, keep in mind that user and site administration have remained in the thick client. Do not try to minimize your investment in user training with the new Studio interface; while it is decidedly cleaner and more usable, some CoreMedia customers say it still can be difficult to master.

Contributing Content

CoreMedia refers to in-context editing as “preview-based editing,” which combines in-page editing with additional functionalities. The contributor can work with content in this mode, as well as edit the web page layout.

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	



Figure 120. In preview-based editing, you can edit in the CoreMedia Studio interface using Studio, or Editor. Note that only Firefox and IE are considered “approved” and are thus the only supported browsers.

Through a partner, CoreMedia can integrate with Microsoft Word and other Office 2010 products via a plugin, which is frequently licensed by government customers. This requires extra licensing. The integration works bi-directionally, enabling the user to upload and download documents to/from the CMS and Word.

For repository search, CoreMedia uses Solr (like many of its competitors). The vendor used to offer Microsoft FAST as well, and claims it still supports it; however, since Microsoft is now pitching that product as part of a SharePoint infrastructure only, you probably should focus on Solr, instead (especially if your infrastructure is not on Windows).

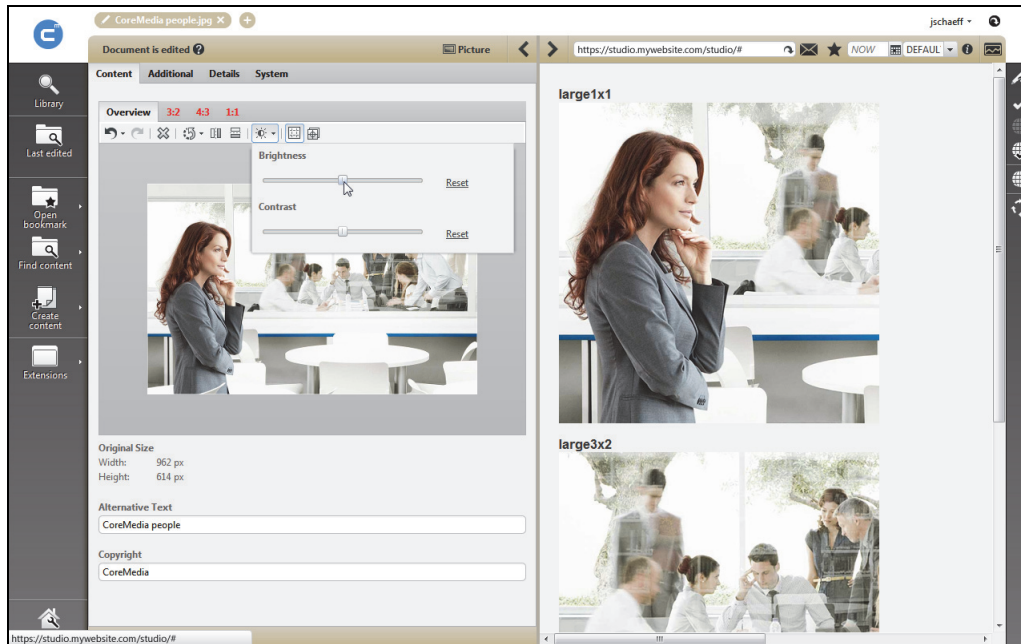


Figure 121. Image editing capabilities in CoreMedia include cropping, rotation, and the adjustment of brightness and contrast.

Content Lifecycle

CoreMedia has taken an interesting approach with regard to workflow. Workflows can easily be created and managed using the browser, as with other vendor's products, but in the background, the actual workflow instances run as separate processes. This means that workflow can run on a separate machine. The advantages of this separation include better load balancing and reliability (the CMS can go down without leaving in-process workflows in an unknown state or requiring a restart of interrupted workflows). The workflow engine can also be further extended using custom coding in Java, e.g., to notify third-party systems of certain events or involve external services in the workflow. A very simple example would be to send an email to an approver, rather than using the editorial client for approval; but you could also fire off a compliance workflow as certain actions complete, or peek and poke data in a remote system.

Localization of output does not seem to be a problem. One customer (a government organization with embassies throughout the world) has deployed sites in 170 languages. For customers of this type who need to translate content into many languages and keep it all in sync, CoreMedia has an SDL Trados integration. (SDL, of course, is also the parent company to SDL, reviewed elsewhere in this report; it seems that SDL's translation subsidiaries continue to play nicely with competing WCXM products.)

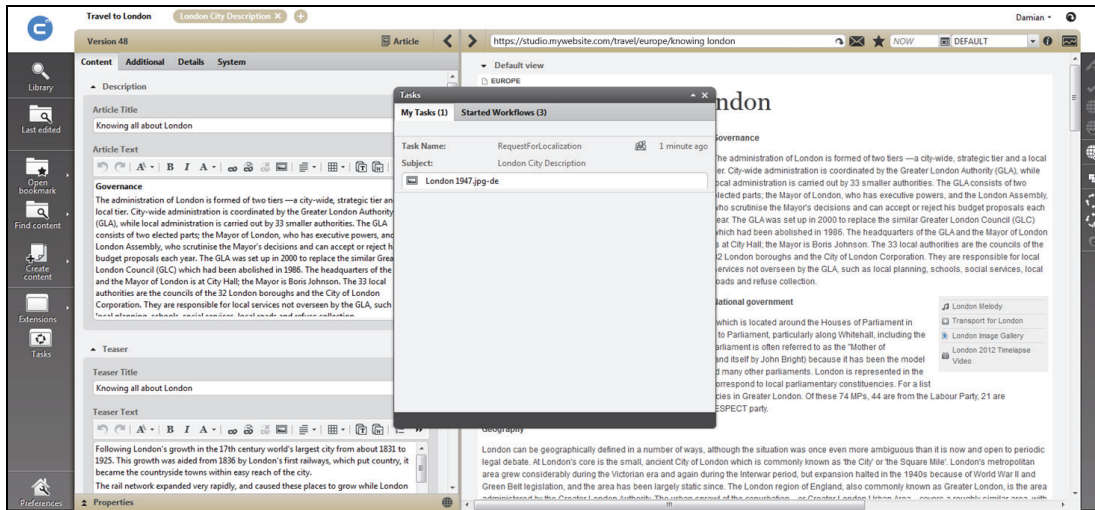


Figure 122. Assigning a task in CoreMedia workflow system from the Studio interface. Note that CoreMedia ships with a series of sample workflows that can be used for simple approval processes.

Experience

Publishing

With a series of European telecommunication companies as customers, CoreMedia provides a comparatively wide array of available formats and targets for content delivery, where content and/or layouts differ depending on the channel to which it is being published.

CoreMedia has various offerings that enable contributors to preview the same content in different formats. This includes various mobile devices, such as Android, iPhone, etc.

Experience	
Publishing	
Standards Adherence	<input type="checkbox"/>
Multichannel	<input type="checkbox"/>
Mobile	<input type="checkbox"/>
Digital Marketing	
Site & Campaign Analytics	<input type="checkbox"/>
Testing & Optimization	<input type="checkbox"/>
Segmentation & Personalization	<input type="checkbox"/>
Social Media Integration	<input type="checkbox"/>
Promotional Campaigns	<input type="checkbox"/>
Community & UCG	<input type="checkbox"/>
Workplace	
Collaboration & Networking	<input type="checkbox"/>
Dashboard	<input type="checkbox"/>
Ancillary	
Site Search	<input type="checkbox"/>
Online Forms	<input type="checkbox"/>
Module Ecosystem	<input type="checkbox"/>

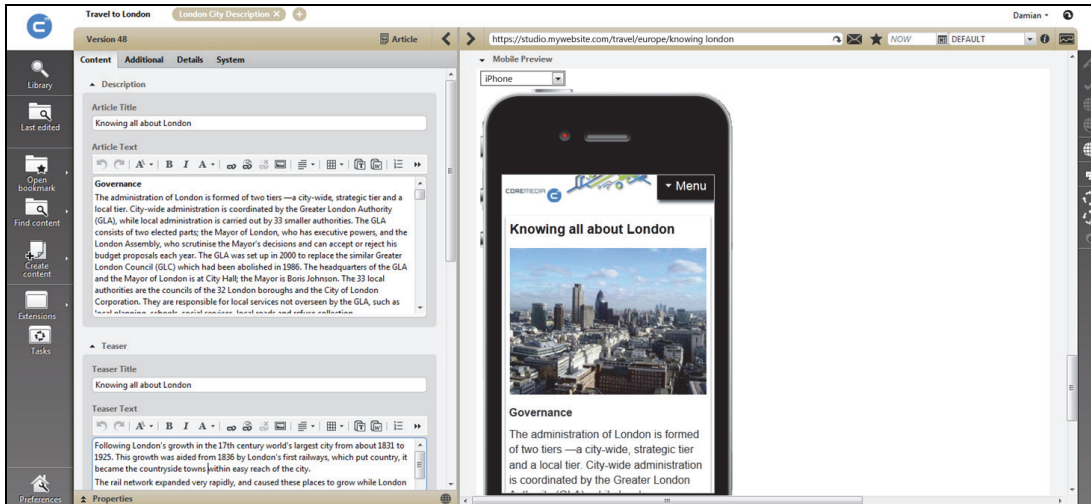


Figure 123. You can preview content from CoreMedia Studio to see how it would display on an iPhone once published.

Digital Marketing

Starting with version 6, CoreMedia introduced a separate environment (the Studio) for marketing and experience management activities. In essence, this is a different, modernized view of some of the previously existing — as well as new — capabilities in CoreMedia.

The Studio is where you conduct marketing-oriented tasks such as personalization and segmentation, social media management, and site analytics.

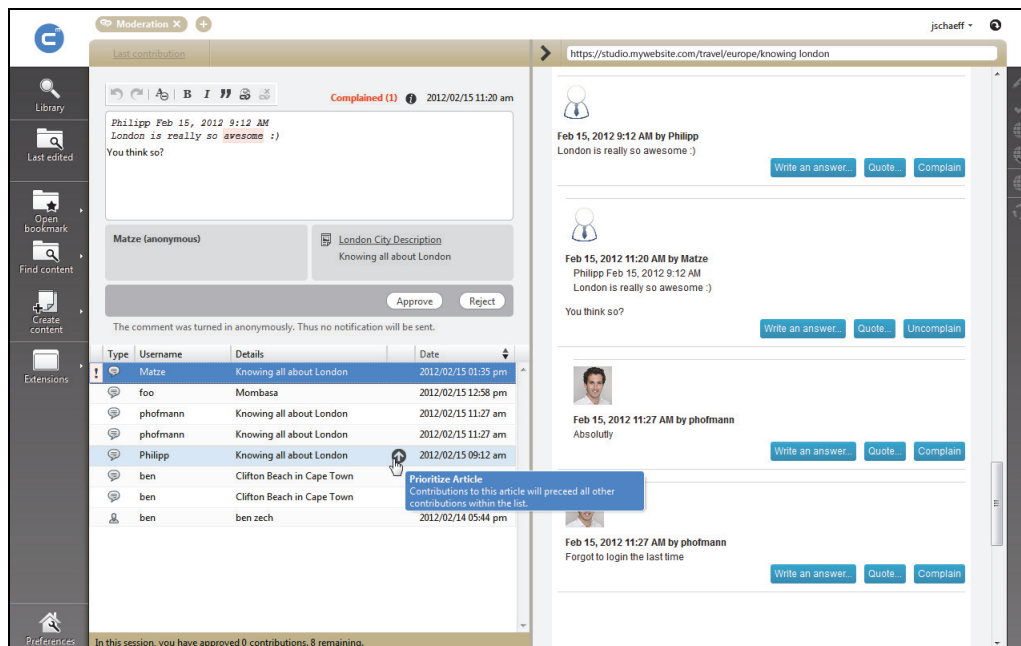


Figure 124. A view into social content moderation from CoreMedia Studio.

CoreMedia’s optional Social Software Extensions package, first released in January 2008, features blogging with ratings, tagging, commenting, trackbacks, feeds, and moderation capability, plus optional user registration via LDAP directory integration. Note that, like in the case with HP, this package deliberately stores user content live on the delivery tier, with no automatic push-through of content into the full CMS, where you could apply other native management services, and devise a longer-term archiving scheme. Nevertheless, the ratings data can be captured into the analytics system and used for driving page optimizations of various kinds.

Among other social components, CoreMedia CM can publish to social networks like Facebook and Twitter. Rules-based personalization is also offered through the Studio interface.

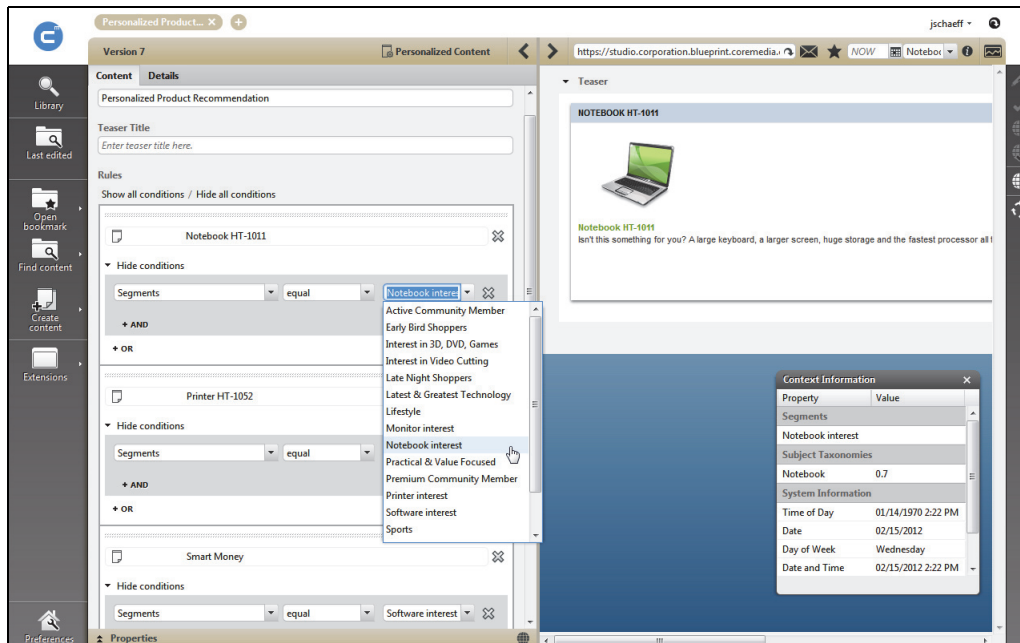


Figure 125. A view of the Studio interface showing segmentation and personalization capabilities

Rules can be created for each content type and can be tied to criteria such as audience segments, categories, price, or popularity. A dialog-based UI allows non-technical users to build rules based on selecting values from drop-down menus, pointing and clicking to establish rules similar to Amazon’s “if you looked at this, you may also like this” Algorithm. You should note that some of CoreMedia’s personalization features come from integration with a German vendor nugg.ad. CoreMedia’s Adaptive Personalization module feeds off nugg.ad’s data on demographics, target groups, and audience insights. The plus side here is that you can tap into your customers’ socio-demographic information — including gender, age, income, education, product interests, and surfing behavior. However in order to take advantage of this data for your own content personalization, you will need to pay two vendors: to CoreMedia for the personalization module, and to nugg.ad as a data-as-a-service provider in a separate contract.

More advanced personalized content delivery routes can be crafted with enough patience and knowledge of your business and your various audiences: both online and on other channels. Given the fact that all processing of real-time data and dynamic generation of personalized content happen on the fly, server-side in the delivery tier of the infrastructure (in the CoreMedia Content Application Engine, “CAE,” to be specific)—, carefully test performance here, as well as how accurately the rules are performing. This is also an important consideration if you look at personalization from the channel-specific perspective. For example, is mobile personalization creating slower page loads? Are mobile audiences getting accurate, location-based, personalized content at particular moments in time?

In terms of analytics, CoreMedia provides an integration with Google Analytics, where — depending on how you have it set up — it can provide a glance into various depths and slices of your data: from simple page views to visitor segments and their behavior. Note that starting in version 6, CoreMedia abandoned its in-house built analytics module with plans to replace it with connectors to third-party tools, such as Omniture.

Ancillary Services

For search, you can apply the CoreMedia search module based on open source Solr for editorial, intra-system and front-end site search.

Vendor Intangibles

CoreMedia is a strong European vendor with a particular focus on websites and extranets for the media and telecommunication verticals (where the majority of its customers reside), as well as (to a lesser extent) government and institutional customers.

The company claims its content management revenue continues to rise. In the past few years, CoreMedia successfully expanded beyond its German base, with 60 percent of its revenue coming from outside of Germany (split more or less evenly between Western Europe and North America). Offices in London, San Francisco, and Singapore support these areas and the rest of the world.

CoreMedia’s licensing model is complex, combining per-CPU fees with concurrent editor seats. CoreMedia charges separately for Content Management servers (\$28K per CPU core) and Content Delivery servers (\$21K per CPU core). Adaptive Personalization and Device Delivery modules cost extra: \$6,500 per Delivery core. CoreMedia Social Software is another optional module that starts at \$13K per core. Developer, preview, and staging servers are included in these licenses. The company cites median deal sizes of around \$150,000-250,000. This seems low, given the pricing above and the comparatively hardware-intensive nature of the platform. Many CoreMedia customers end up paying much more — upwards of \$1 million.

According to CoreMedia, many customers start with five CPUs. Portal integration is separately licensed. As a buyer, you should carefully evaluate your options and insist on well-defined, concrete licensing terms, so you can compare “apples to apples” with other vendors. Training is available (for a hefty fee), but customers laud its quality.

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

Most CoreMedia projects are implemented through partners with assistance from CoreMedia consultants. On occasional projects (with new technology or when the customer insists), CoreMedia handles the project with its own professional services group. CoreMedia also has a partner certification program so that partners can certify their knowledge with one or more exams up to different levels. A Platinum Partner (the highest level) must have a number of certified engineers, as well as several completed projects using the latest major release. CoreMedia's partner program is reasonably ahead of most vendors in this regard.

In addition to its staff in Hamburg, CoreMedia can now offer some support from a comparatively small number (21) of employees in the US, though usually you will still deal with German contacts. To be fair, we have heard of few issues with this from customers in other time zones, who are generally impressed with the knowledge, ability, and responsiveness of CoreMedia staff.

The company maintains a slow but steady release schedule, favoring stability (much like German competitor FirstSpirit, and probably likewise influenced by giant SAP's habits, to which large German customers have grown accustomed).

Conclusion

If your enterprise does not fall into one of CoreMedia's core media or telco verticals, it still might be a suitable choice for a fairly large public website. On the other hand, if your project is focused on building a Digital Workplace site, then CoreMedia is missing quite a few features (most notably document collaboration), compared to other vendors like IBM, which bring a stronger intranet focus. If you're running a TV station, you'll definitely want to have CoreMedia on your long list — if you can afford the millions it may require to implement it.

Despite dipping into overseas markets, CoreMedia remains a very European-focused vendor. It does provide user interfaces in various languages, but has not (yet) gained a significant footing in regions such as France, Spain, or Italy.

This is a complex — but scalable — system that will please most Java developers and fit well in Java-centric IT organizations, especially if in-house developers already speak JSF and are comfortable with Eclipse and Maven. This is certainly not a simple system, but if you're serious about this class of CMS solution, then ideally you'll have an IT or internal R&D organization that is equipped to handle it. CoreMedia is not an entry-level or "starter" CMS. It's a "toolbox," as some customers say. Bring ample financial and human resources to the table if you want to take full advantage of CoreMedia's breadth of functionality.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Percussion: CM System

www.percussion.com

Vendor at a Glance

Specsheet	Percussion: CM System 6.7
Geography	Global, with emphasis on North America and the UK
What's New	<ul style="list-style-type: none"> • Introduction of an additional product called CM1 as a simpler alternative to its traditional CM offering
Strengths	<ul style="list-style-type: none"> • Mature Java framework is developer friendly, with extensive APIs, and decent standards support • Very strong native support for component content reuse, and CMS user interface can be set up to follow accessibility standards • “Active Assembly” can give power users substantial control over page development and layout • Decoupled architecture is a very good fit for many enterprise scenarios with custom or heterogeneous website delivery applications • Percussion is a privately-owned, pure-play WCM vendor with no sideline businesses or “parent corporation” to distract them • CM1 product provides an alternative for simpler scenarios
Weaknesses	<ul style="list-style-type: none"> • This is a tech-heavy platform: most implementations will require custom Java development; templating requires a developer • Admin interface has always been rather clunky, and the contributor interface tends to cater to power users • Personalization, analytics, and UGC features bring ample added costs • Workflow system can feel rigid and unnecessarily complicated • Customers have said that sales, support, and training can be erratic • Third-party specialists and integrators with Percussion experience can be comparatively very hard to find • Company traditionally has tended to favor power-user features and architectural sophistication over ease of use • CM1 is a separate product that could distract or divide company efforts • Company has experienced some leadership turnover, which may bode poorly in future performance
Potential Fit	Ultra-Large Single Site, Global Enterprise, Multichannel Publishing
Unlikely Fit	All simpler scenarios, Advanced Marketing Portal
Compare To	HP, SDL, Oracle, Adobe, OpenText WEM
Operating Systems	Windows, Solaris, Red Hat Linux
Repository	Databases: Oracle, MS SQL Server, IBM DB2
Client	All major browsers on Windows, Firefox on Mac. Administrative UI uses applet technology, as does the rich text editor
App Platform	Java/J2EE — bundled with JBoss

Specsheet	Percussion: CM System 6.7
Licensing	Server based. Median deal size roughly US\$150,000
Ownership	Privately held

Summary

Like many vendors, Percussion has put a great deal of effort into finding ways to differentiate itself in a crowded market. The company emerged from identity-crisis mode late in 2008, with a vision of giving web managers, digital marketers and content chiefs tools for updating and/or repurposing online content *quickly* in response to business needs that might change day to day (or even hour by hour).

The product formerly known as Rhythmyx (a moniker that has officially been retired in favor of “CM System”) has acquired a bevy of features geared toward lead generation, community building, brand loyalty, customer retention, intelligence gathering, dynamic personalization of content, identification of market trends, and so on. This product caters to the needs of power users — or should we say, “entitlement-rich” users with the kind of broad power (and system knowledge) required to publish and repurpose content on a moment’s notice. Similarly, casual contributors or those accessing the product’s traditional interfaces may encounter a clunky, techie-feeling system.

Percussion expects a certain degree of sophistication not only on the part of administrators and contributors, but also developers. As with Oracle, Percussion’s product is platform-like and achieves its dynamism by virtue of a multifaceted, abstract content model (designed with flexible page-assembly in mind), rather than through deft use of XSL.

It’s an interesting approach and worth considering if you’re in the market for a WCM platform that offers better than average globalization and localization potential, with a decoupled architecture. You’ll want to bring at least \$150,000 with you to get started, and you will need help from experienced Java developers to get things rolling. This product tends to be selected when IT people have a larger say in decision-making.

Although CM System is certainly capable of handling simpler scenarios, its price point and architectural sophistication make the product overkill for many of these scenarios. This is a serious Java-based platform for managing structured, reusable content on a large scale — with development and administration requirements to match.

To cater to simpler scenarios, Percussion has launched another product called CM1. Although it works off the CM codebase, you should not consider it a “starter” package, since you have to

Scenario Fits	
Simpler Site	
Informational	<input type="radio"/>
Microsites & Landing	<input checked="" type="radio"/>
Mid-Range	
Basic Digital Marketing	<input type="radio"/>
Mobile Site	<input checked="" type="radio"/>
Community Oriented	<input type="radio"/>
Complex Site	
Advanced Marketing Portal	<input type="radio"/>
Global Enterprise	<input checked="" type="radio"/>
Multichannel Publishing	<input checked="" type="radio"/>
Ultra-Large Single	<input checked="" type="radio"/>
Digital Workplace	
SMB/Departmental	<input type="radio"/>
Global/Enterprise	<input checked="" type="radio"/>

run it separately from the main CM product. Additionally, do not assume that there's a clear migration path from CM1 to CM, or vice-versa; you'll want to commit to one or the other. CM1 is akin to a cheap version of a luxury automotive brand; we advise that you wait-and-see on this one.

Introduction

Founded in 1994, Percussion Software (based in Woburn, Massachusetts, USA) began life developing add-on tools for Lotus Notes and Domino. The company developed its Rhythmyx CMS product in 1999, and in 2007, Percussion spun off its Lotus Notes integration group as a new company called Axceler. Since then, the 60-person company has focused entirely on web content management, with no sideline businesses.

The Rhythmyx product — renamed Percussion Content Management System (or just Percussion CM System) in 2009 — was originally modeled on a “component” view of WCM, as opposed to a page-centric approach, and it has historically been best suited for environments where content is highly structured.

It's worth noting that many of CM System's more interesting personalization and analytics-driven features require the purchase of additional modules (at \$30K to \$50K each). There are three main modules (which Percussion calls “Solutions”):

- **Personalization Solution** — Has features that allow marketers to match content to individual users based on demographic, psychographic, or behavioral segmentation models automatically
- **Community Marketing Solution** — Captures user ratings and recommendations, and facilitates the creation and management of blogs and feeds
- **Web Analytics Solution** — Provides a variety of traffic-analysis reports and overlays, and some analytics-driven automation around tagging and things like “Most Viewed” lists; also allows direct navigation into system content from analytics views so that content can be revised quickly in accordance with real-time changes in visitor preferences or traffic patterns

Together, these modules provide a powerful toolset for managing highly dynamic sites, and for many customers, the value proposition of a Percussion system rests mostly with these modules rather than the CMS itself. That's not to say the core CMS doesn't have its own differentiators, but you should be clear on what your needs are, and when you see a Percussion demo, be sure you understand what you're seeing. Ask to see the core CMS first, and view the solutions separately.

Percussion recently has built a separate, more productized offering called CM1. CM1 uses the same code base with a simplified user interface. Percussion says it is targeted for people using products like WordPress and are moving from blogs to simpler sites. CM1 provides an AJAX-/widget-based front-end, with drag-and-drop capabilities to develop small sites quickly. It is a less expensive alternative, but it has limited extensibility options.

Technology

Technical Administration and Security

Percussion speaks to a particular organizational paradigm where the licensee has divided content management and site/interaction management into two separate concerns, perhaps even managed by separate teams. Percussion argues that by separating content management cleanly from site management, a licensee is freed to use arbitrary flavors of code (ASPX, JSP, PHP, CF, and others) on delivery servers, to assemble dynamic content — and manage user interaction — without the CMS unduly influencing the choice of page-delivery technology.

This is true in many instances. In some cases, however, an organization might want to empower content managers to control content delivery and interactivity at a more intimate level, arguing for a more coupled architecture that has the content management software deliver content, support micro-applications, and generally manage the website, as well. Percussion is only now beginning to feel its way toward this latter model of web publishing. In Version 6.6, Percussion created a “Publish Now” command, for example, that allows an appropriately permissioned individual to publish a content item straight to the delivery tier, right from the in-context editing environment. This allows contributors *and* site designers to do edits and page optimizations (and see the effect on web traffic) with minimal turnaround time. However, it also puts additional focus on the need for good role-management discipline, version-control discipline, and administrative insight in terms of (for example) audit trails. The power to do rapid optimizations is a good thing if you’re a marketer with wide-ranging CM needs, but it comes at a cost.

Percussion is a Java-based system that runs in the (bundled) JBoss application server. It makes use of a number of open-source pieces, including Hibernate, Spring, and MyFaces (an implementation of JSF).

At an implementation level, the architecture is quite sophisticated and you may find it overkill for simpler applications, particularly those that feature uncomplicated pages that contain mostly fully baked, unstructured content. On the plus side, the system is standards-based (even the more obscure pieces build on openly published industry standards), with no reliance on arcane one-off scripting languages (e.g., Oracle/Stellent “Idoc Script”) or baroque proprietary APIs that ultimately get superseded by yet more baroque APIs. Percussion’s standards orientation tends to mitigate the system’s complexity somewhat, inasmuch as developers with a solid understanding of Java, Velocity, and JEXL can start to become productive early on, without having to spend large amounts of time in the classroom learning arcane technologies. Nevertheless, the content model and system architecture do come with a learning curve. You can either pay for the required training, or hunt down experienced Percussion experts. On-the-job training is not going to be the cheapest option.

Roughly half of Percussion’s customers who deliver content dynamically (as opposed to baking-style architectures) use CM System to publish into a delivery tier that pre-dates or was

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

developed independently from their CM System implementation. The other half builds a new delivery tier with their CMS implementation. Percussion encourages licensees to utilize open-source Java web application frameworks and scripting languages to develop presentation tiers that interact with structured content published by their management tier. Content logic and bindings are built on the CM System side and then executed at the delivery tier.

Unlike many Java-based tools in this report, Percussion can point to .NET as a popular delivery environment among its clientele. JSP technology is still the most common option, however, and Percussion is one of few vendors in this report (aside from Adobe and Refresh) that can claim to support JSR 170. Note that Percussion’s implementation covers only the query portions of the standard, however — not the “update” API.

The system is heavily XML-oriented, relying on a combination of Content Type definitions, Content Items that meet those Type definitions, and Velocity templates to separate content and presentation, while also allowing a high degree of page compositionality. Although the system uses a database to store content, the elements are represented behind the scenes as JCR nodes.

Version 6.7 heralded the arrival of a package-based modularization paradigm. This wasn’t really a product architecture change so much as new tooling to allow existing functionalities — as well as extensions written by consultants and customers — to be hived off into reusable components. The components can be mixed, matched, shared, and maintained separately from each other and from the rest of the system. Percussion hopes to build a library of packages contributed by the company, customers, and consultants, from which Percussion customers can draw, but there are no public packages whatsoever. For now, it’s a case of roll your own and (re)use it yourself.

One of the custom packages Percussion likes to demo is a Twitter integration piece that features an applet that lets you create tweets from within the CMS. Because you’re working in the CMS, you can use the system’s Search features to find links to related site content (and apply a URL-shortener as needed), workflow your tweet, find similar tweets that have been posted and archived, and so on. If company policy on tweeting changes, the Twitter-integration piece can be modified and redeployed easily, by itself, without disturbing any other parts of the system.

Package Status	Config Status	Name	Publisher	Version	Description
<input type="checkbox"/>	✓	perc.listbox	Percussion	1.0.6	Display multi-level lists as tabs, expanding accordions...
<input type="checkbox"/>	✓	perc.calendar	Percussion	2.3.0	Presents event items in a configurable calendar forma...
<input type="checkbox"/>	✓	nava.flashgallery	Navigation Arts	2.0.0	Customizable flash image gallery with control panel a...
<input type="checkbox"/>	✓	mlcr.sxworkflow	Molecular	2.4.0	Three-step governance workflow for Percussion CM S...
<input type="checkbox"/>	✓	redr.donationcentre	Percussion	1.0.0	Adds a donation widget, secure payment engine, don...
<input type="checkbox"/>	✓	auto.bulkfeedimporter	AutoTrader.com	2.1.4	Enables bulk importing of RSS feeds as content items...
<input type="checkbox"/>	✓	vlon.eventpicker	Visit London	1.0.3	Widget provides "find an event near you" functionality...

Figure 126. The Package Manager allows you to install or uninstall arbitrary pieces of system functionality individually, giving CM System a “mashup” configuration architecture.

Percussion’s other product, the newly launched CM1, is architecturally similar to CM System. However, unlike CM System, CM1 comes with an embedded database and supports only Tomcat. It also has limited capabilities for extensibility and integration with external applications.

With respect to the publishing model, the product is essentially a decoupled “baking” or “parbaking” system. That is, the system pre-generates static HTML or XML snippets, documents, or database records when content is approved. You then deploy that content to the delivery environment of your choice for consumption by site visitors.

CM System can provide some flexibility here. The product can publish structured content to a relational database, or it can pre-render HTML content blocks (for better performance on the delivery tier). As with OpenText, Percussion can apply multiple different transforms to the same content object for different environments, which puts less stress on having to revise delivery applications as content models change. Your delivery system can:

- Serve up static HTML pages that were pre-generated in CM System and promoted to the web server
- Assemble HTML snippets by transforming XML content promoted from CM System
- Read from a database into which CM System publishes

Your delivery application can also execute Percussion-generated dynamic server pages (JSP, ASP, ASP.NET, PHP).

For basic pushes, CM supports FTP, as well as Secure FTP natively. Incremental deployments (either scheduled or in response to “Publish Now” events) are handled by native CM Server code. Transactional deployments can bring any number of different hiccups, and Percussion has experienced a few of its own. For example, if an incremental deployment fails, it can hold up all subsequent deployments until the problem is addressed by an engineer. However, dependencies are checked prior to an incremental deployment taking place, so that referential integrity is never compromised on the delivery side.

Note that the publishing subsystem can be broken out as a separate service, although some customers run it on the same box as the CMS. Those with large or particularly active sites with many content transformations may want to split the publishing service off to a different machine entirely. Of course, this has licensing implications.

As with any baking system, you’ll want to measure the time it takes to publish a typically sized edition for deployment. This becomes an issue particularly if you embed a lot of business logic that needs to be resolved at publish time. Percussion notes that the “big bang” style of once- or twice-daily site builds is not as prevalent with customers these days as it once was. Incremental real-time publishing is becoming more popular. One advantage to the full-on, bake-and-push approach, however, is that you can roll back an entire site to a previous edition (a known good state), if necessary.

The CM System rights model takes a bit of getting used to. In Percussion, a Role is a group of users with the same privileges, but above the level of Roles is what is called a Community. A Community represents a collection of Roles that require access to the same information in CM System. A Role can map to more than one Community, and users belong to all of the Communities associated with their Roles. Thus, when a user logs in to the CMS, the system only displays components associated with the given user’s Communities.

The system is flexible enough that you can manage roles in your LDAP directory, if you don’t manage them in the CMS itself. It’s also possible to write your own custom role handlers.

The Role and Community metaphors allow for flexibility in mapping permissions to various stages in a workflow, as well as for achieving personalization of contributor interfaces. However, it’s an area you’ll want to spend ample time investigating from standpoints of capability, complexity, and performance. In typical Percussion fashion, the system’s rights model is abstract, elegant, and powerful — but not something you’ll wrap your head around in a day.

CM System provides some limited — though useful — reporting. Editors can run an “impact analysis” of dependencies among content objects, and see what the implications are for making a change to any discrete object. For example, if you change an image, what will happen on all the associated pages where that image should display?

The product ships with some Excel-formatted reports for content inventory auditing, along with various status reports, including a listing of “stale” content. On the plus side, Excel data is queryable; on the downside, it’s a very poor interface if you actually want to act on the reports. Most other tools with reporting systems in this report provide clickable results, so that you can easily redress problems like orphaned workflows.

Development

The product’s SDK is more extensive than most. It features an Eclipse plugin that can connect to the repository, making development of templates, code, and Web Services much easier. (Figure 127 shows the WSDL-viewer’s diagrammatic overview of the API.) Code examples are part of the SDK as well. Developers will appreciate the fact that, as with TERMINALFOUR, JavaDoc is available for essentially all of the product’s core classes.

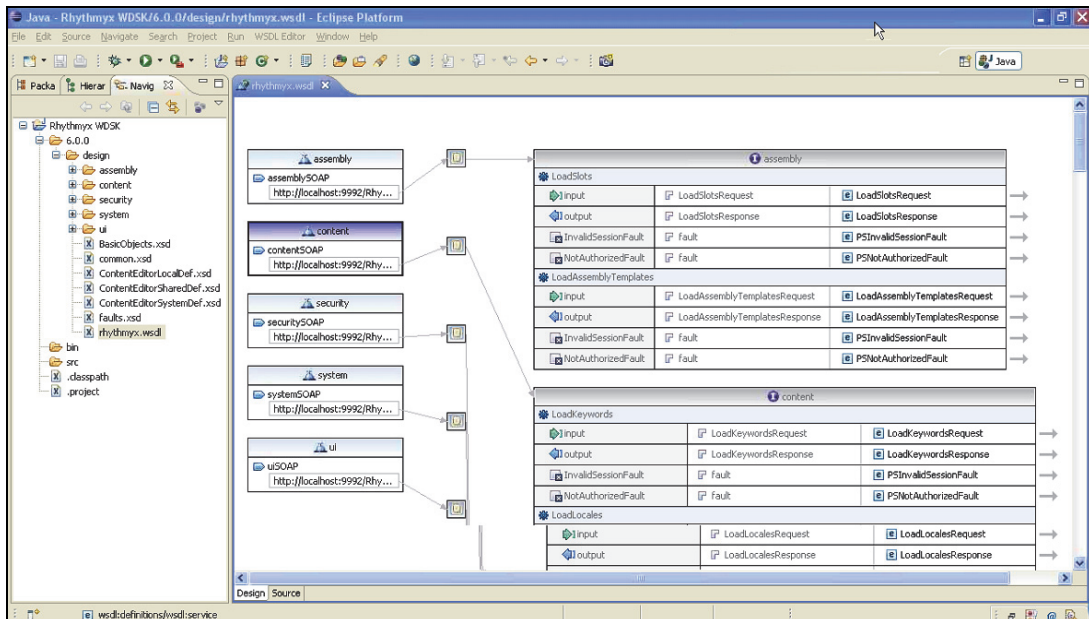


Figure 127. The SDK comes with an Eclipse plugin. Here, the WSDL viewer gives a diagrammatic depiction of key portions of the product’s Web Service API.

CM System can be extended with custom JSPs, servlets, and/or Spring beans, and there's a useful Web Services API as well as the JSR 170 API. In general, the product's customization and integration potential have to be considered good.

For templating, the product applies the Velocity templating engine (an Apache project), using JEXL — Java Expression Language — for additional scripting flexibility. Although CM System can also apply XSL (in case you have a large library of preexisting XSLT, say) and supports XSL-FO for PDF output, all the templating code in the product is Velocity-based by default. In addition to performing two to three times faster than XSL processing (hence the name), Velocity has a significantly shallower learning curve for developers than most other templating systems.

One of the main concepts behind page rendering here is “Active Assembly.” You can think of a web page as a collection of components, with each component possibly containing other components (that contain other components, and so on). When a page designer or other power user opens a Content Item and drops another Content Item into one of the first item's “slots,” the two items become linked by an Active Assembly Relationship.

At publish time (or when the Content Item is previewed), Percussion's CM System automatically executes the Active Assembly Relationships in a process called “recursive rollup.” The most deeply embedded Content Items are transformed or formatted first; then the Content Items that include them are formatted; then the Content Items that include those Content Items are formatted; and so on, until the top-level item that includes and displays all the others is formatted.

As you can see, this architecture takes some getting used to, and formal training of your developers is necessary. Where other systems might hard-code nesting relationships in the actual templates, CM System takes a more dynamic approach, based on a combination of design-time and runtime dependency injection. For someone accustomed to working with straight XSLT, it's bound to feel unfamiliar at first.

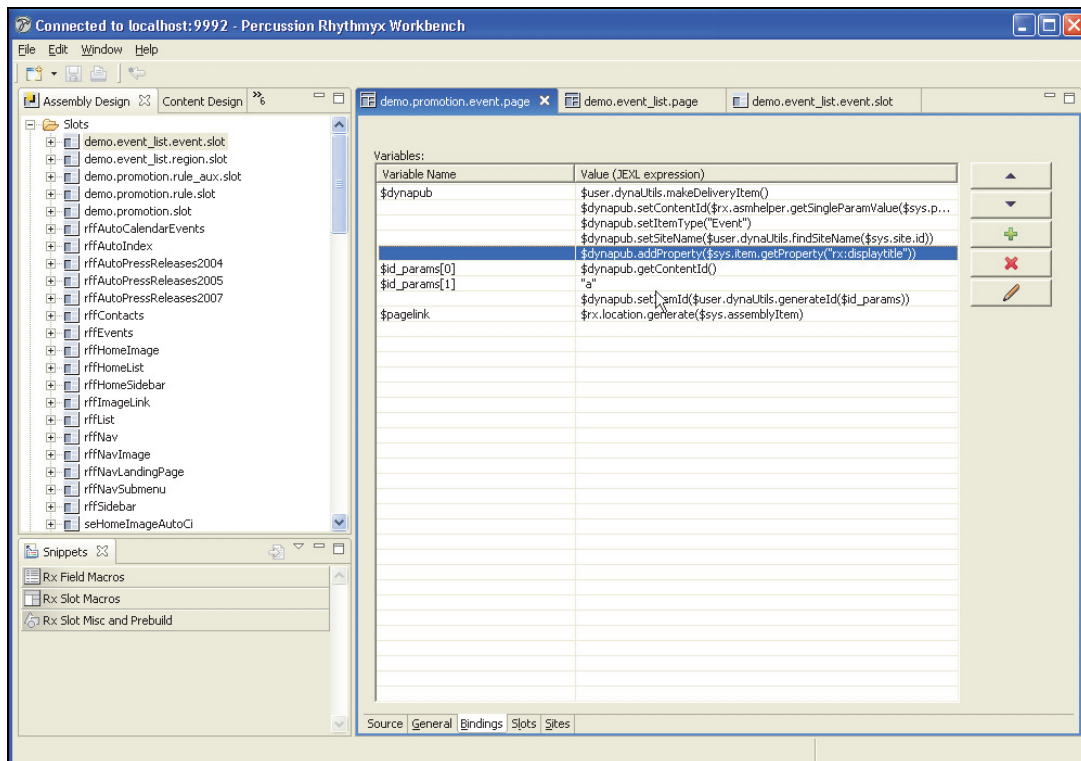


Figure 128. Using the Developer Workbench to edit variable bindings available in a "slot."

Performance

After longstanding customer complaints about editorial performance, Percussion says the latest point releases of the product have seen increases in the speed of both the Active Assembly and Explorer interfaces, along with 37 percent faster publishing performance.

You'll want to test the admin and contributor interfaces yourself carefully, to see if they meet your performance expectations from a responsiveness point of view. Note that both clients rely on a combination of DHTML and Java applets.

On the delivery side, performance is on a par with other decoupled, baking or par-baking systems and is subject to the same considerations with regard to caching, use of CDNs, page complexity, and ratio of static content to dynamic content.

Content

Contributor Experience

Overall usability isn't necessarily poor, but the admin and contributor UIs are menu- and dialog-heavy, and many areas of functionality (including the WYSIWYG editor and the entire admin client itself) are implemented as Java applets. The contributor UI does a good job of hiding complexity from the user, and you can exercise some control over this by making various pieces role-restricted as to visibility. Make no mistake; this is a power user's product, and the contributor UI is a power user's UI.

Like several other tools at this tier, Percussion has not one, but two interfaces: the so-called "Explorer" interface (a Java application, implemented as an applet), and the newer "Active Assembly" interface. The AJAX-based Active Assembly UI is intended to be the primary user interface for most content contributors. Percussion targets the "thicker," more powerful Explorer interface toward power users and administrators who understand the organization of the repository and don't mind giving up simplicity for the additional features that the Explorer interface provides.

The Explorer interface, which was rewritten as a Java applet in V5.0, came under customer scrutiny because of poor load times and latency issues related to passing data back and forth to the server. The performance improved somewhat since then, and with Version 6.7, you may not need to visit Explorer very often (unless you're an administrator or need to touch deep-config settings), because a great number of features have been moved from Explorer to the Active Assembly interface now.

Given the product's past performance issues and the many recent changes to the Active Assembly client (and the intensive use of applets), we strongly urge you to test both interfaces (Active Assembly, and Explorer) carefully with real contributors — under combat conditions — before licensing this product.

Percussion does comparatively well with localization. Unlike many vendors at this level (even European ones), who have found their own solution to localization, Percussion has wisely opted for an industry standard called TMX. Translation Memory eXchange is an open XML standard for the exchange of "translation memory." It greatly facilitates localizing the user interface and can be leveraged by developers to provide the product UI in a customer's local language. Percussion's Active Assembly interface also incorporates side-by-side content comparison tools, and workflows can accommodate parallel approvals and synchronized staging of local variants of content.

Percussion has traditionally had a strong focus on accessibility, and Section 508 compliance in particular, to keep a toehold in the US federal market. The Ephox editor has some useful

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

features in that regard, including the ability to apply formatting via keystrokes, along with a nifty validator for content input via the rich text editor against a variety of accessibility standards.

Contributing Content

Percussion has standardized on the Ephox EditLive! rich text editor for WYSIWYG editing, and (as with Oracle UCM) you’ll sometimes see multiple instances of it in a single window if the context item you’re working on has multiple editable fields. The Ephox editor is a Java applet, which is not necessarily a problem, but it does mean potential help desk issues if you have many content contributors, with multiple JRE versions, and multiple browser types, on multiple operating systems.

The system supports copying (and transforming) text from Microsoft Office formats into structured content assets, but the translation from Word to proper HTML is notoriously difficult to get right, and you’ll want to do some investigating here if this feature is something you’ll use a lot.

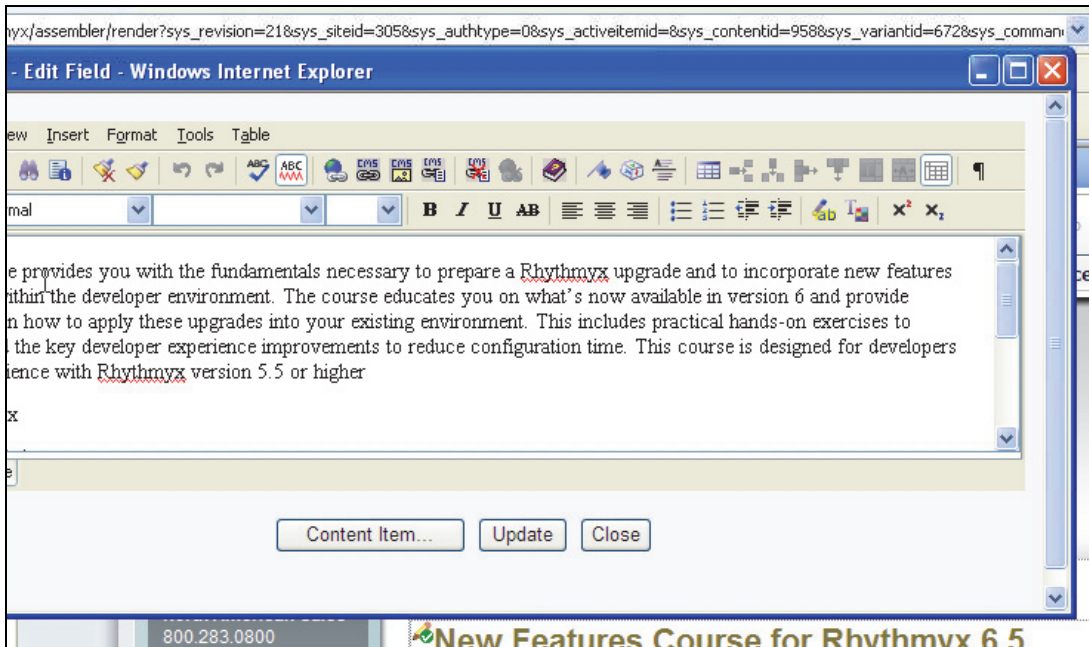


Figure 129. In the in-context interface, a tree control on the left enables the contributor to navigate through the page components by name. The WYSIWYG editor (Ephox) is presented in a pop-up, but edits display as AJAX refreshes in the underlying page. Note that the deprecated Rhythmyx product name still shows up in certain parts of the UI.

Business users get a list-builder interface that is intricate, but lets you build very complex queries (under the covers) without knowing any query syntax. Once the list is created, AJAX drag-and-drop functionality enables users to order and place items in a slot on the page.

The in-context Active Assembly interface is well-suited for content owners who want detailed control over what visitors see. If you’re wondering how this works in practice for dynamic pages (recall that Percussion is a decoupled system without a delivery engine), well . . . you pretty much fake it. Similar to the OpenText approach, using the Active Assembly interface in

Percussion requires presentation templates that mimic the separate delivery tier, but they are just used for editing. Of course, unless actively managed, these templates will become out of date when the actual website is updated. In the end, it becomes your choice whether the Active Assembly interface needs to mimic the site fully. Most editors will certainly want this, and that can mean ongoing developer attention, if your templates frequently get stale. This is a system in which the default assumption is that your templates, once created, will be reused indefinitely rather than being replaced often, because of the compositional flexibility of the system. Nevertheless, it bears investigating.

In terms of accurate previewing, you could also develop alternative, scaled-down contributor interfaces in JSP (using the Percussion Java API) or other technologies (using the Web services API), although doing so could represent a good deal of development. A more practical alternative is to have contributors use the traditional Explorer interface, which — careful here — they may not have seen yet during sales demos.

CM System supports managing content at different levels of granularity, ranging from structured reusable content components to simple, unstructured pages. There is also strong support for building collections of assets (called lists), and placing them in slots on pages. Developers create slots in the Workbench and create default queries to populate the contents.

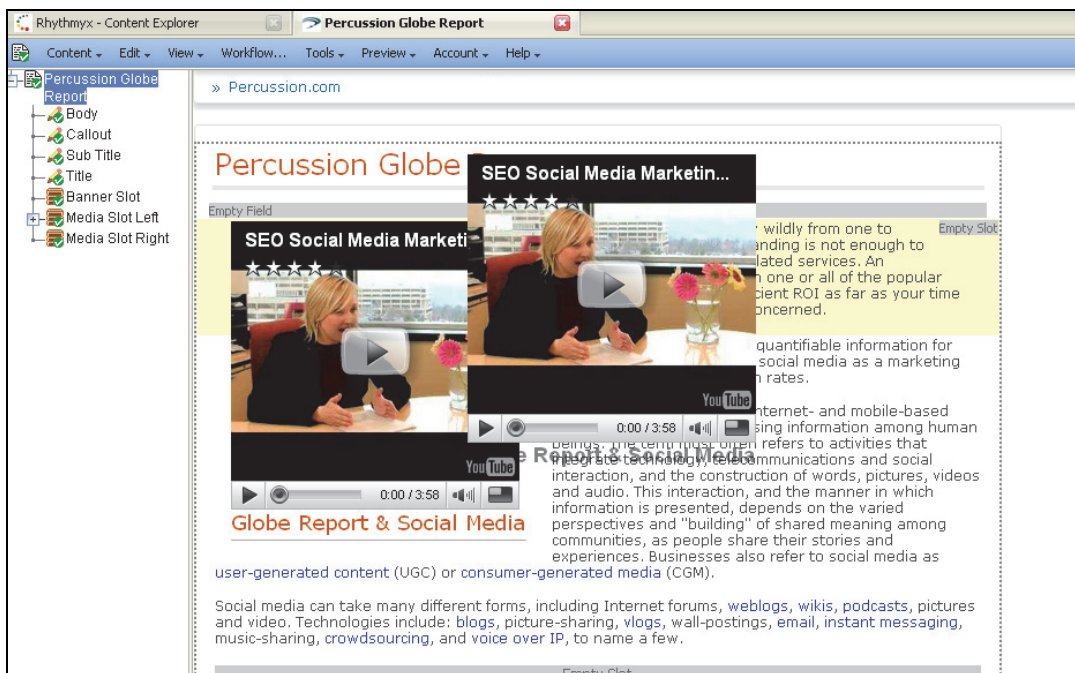


Figure 130. You can (re)order items in a slot by dragging and dropping.

Unlike many offerings that claim to emphasize content reuse (but they are actually page oriented, allowing *repurposing* of page content), CM System is designed with granular reuse in mind. The product’s general ability to apply complex metadata constructs (and inter-object relationships) to various content objects, expose those content bits to content managers at different states, and keep everything in sync automatically, is a key differentiator for CM System within this class of vendors.

CM System has some useful control and reporting on dependent relationships — the kind you’d typically see in “single-source” content management systems (such as Vasont and

Astoria) — that attempt to manage XML components at a granular level for multichannel publishing in technical documentation projects. However, Percussion’s implementation has somewhat less fine-grained control over node-level assembly. The company has since folded this functionality into its main CMS and does not seem to be developing it much further.

Version control is a bit quirky. Automatic (silent) check-in and check-out of content isn’t a built-in feature, so appropriately permissioned individuals “own” their content, manage it themselves, and only rarely need to have others check in and check out the same content item more than once. For many marketing environments, where one person “owns” a collection of materials and is responsible for their full lifecycle, this may work fine, but in other scenarios, it will become onerous to have to check things in and out upon edit manually, or at various steps in a workflow.

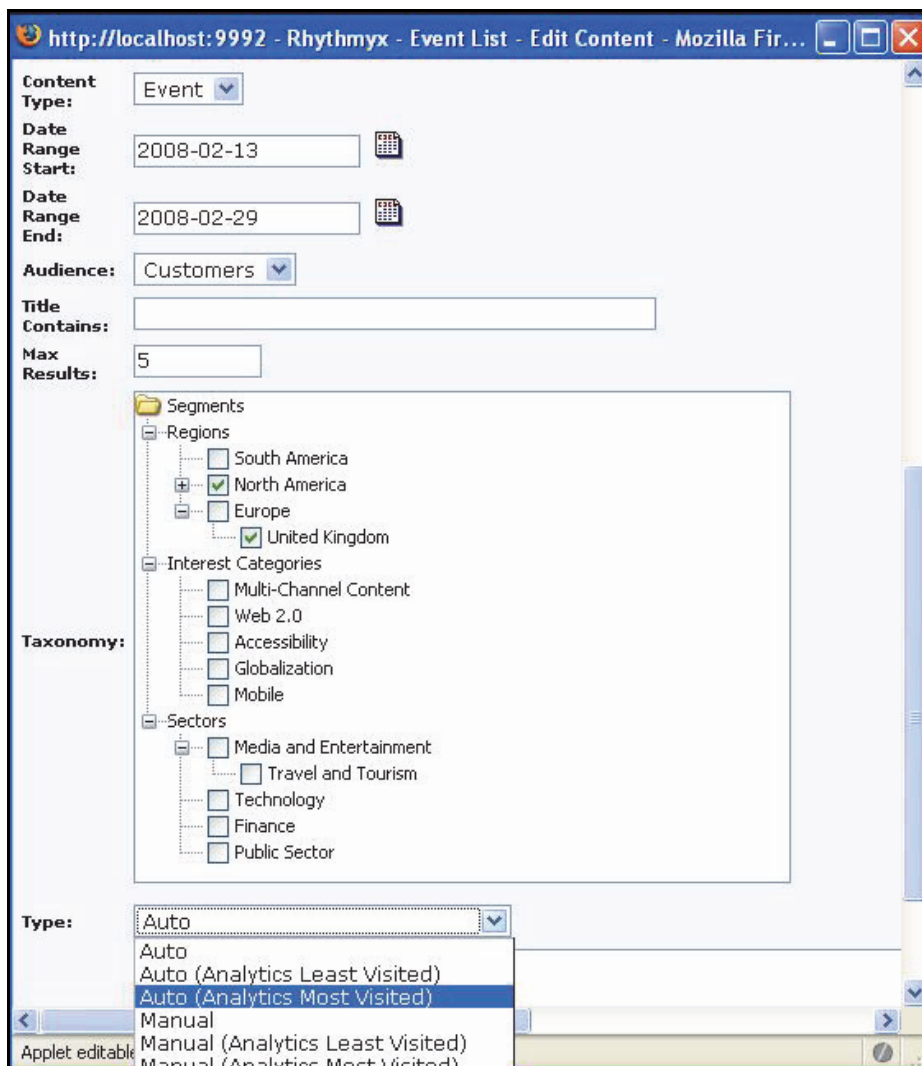


Figure 131. Business users can create “lists” (in the above case, an Event list) that can be slotted on a page, using a self-explanatory UI. There is no need to know a query syntax, or use a pre-canned query to build the list.

Until recently, the product offered few digital asset management features. The product now comes with a lightweight DAM module. CM System can now handle binary content, and has support for WebDAV (to allow users to drag assets into the repository). Percussion also added an asset library and the ability to edit images in the browser. As a result, the system can now easily handle images, Office documents, PDFs, and other formats beyond HTML. It's no substitute for a full-blown DAM system of the type we review in our [Digital and Media Asset Management Report](#), but at least Percussion is no longer feature bare in this area.

Percussion's CM1 has a completely different user interface for content contributors. It offers drag-and-drop templating, and uses AJAX widgets to display content. It enables faster speed to market, but as a result, it is less extensible.

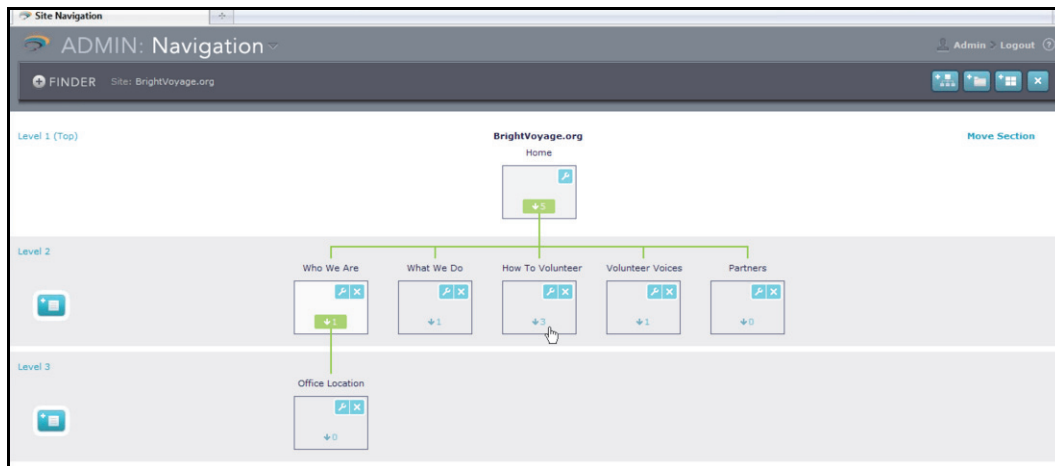


Figure 132. CM1 has a different UI using the same code base.

Content Lifecycle

The system takes a state-transition approach to workflow (Figure 133). Under this approach, content always exists with a given status. In a particular state, people with suitable privileges can do things to the item before it is transitioned to its next state. Many architects find a state-transition approach more intuitive for workflow modeling, although it doesn't handle branching or joins (in other words, parallel flows) very cleanly, and end users typically don't find it as easy to validate visually. Moreover, some Percussion customers find this particular implementation somewhat rigid and difficult to use.

Nevertheless, this kind of state machine seems to work well in environments where there are groups of users sharing the same role and capabilities, or working collaboratively on a document in any given state, as opposed to a linear, task-based workflow that you'll find in many other content management systems. Percussion asserts that its customers typically just want "simple" workflows. That's probably true. It also says, though, that if your needs encompass more robust sorts of process flows that can be mapped to a flow-graph, you may need to look at other systems. Take time to understand your needs well, before evaluating Percussion's workflow.

Percussion bundles a visual editor to enable business managers to make changes to workflows. They can do this in real time, with changes applied to all content in affected states. The system then pops up a state-transition screen (which may or may not confuse the average user).

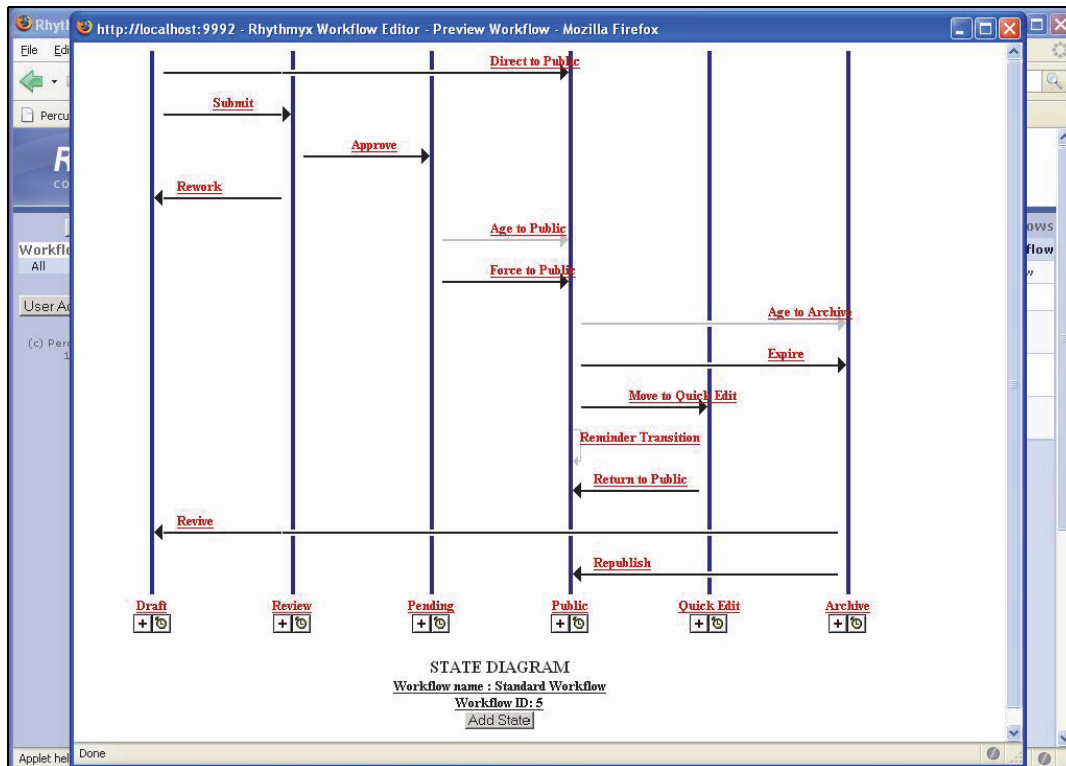


Figure 133. Percussion uses a state-transition workflow model.

CM System provides a capability that allows a user to establish “Intelligent Relationships” across multilingual content. Thus, to create a press release in English and Spanish, the author of the English version initiates a “translate” workflow step, and then CM System generates a copy of the release and establishes a Relationship between the two versions. Next, the system sends the Spanish translator an email notification that contains a link to the copy that requires translation. CM System maintains the Relationship between the English and Spanish versions throughout the content’s lifecycle. The translator can be notified of any updates to the English version, for instance. This is particularly important for sites where language parity is a legal requirement. (On Canadian government sites, for example, the English version of some content cannot go live until the corresponding French version is also approved.)

Percussion offers a nice “cloning” tool that enables users to clone a site, sub-area, or community, for reuse on another project easily. Compared to OpenText WEM (which also has cloning), Percussion seems to offer a bit more functionality (Figure 134), including native facilities for maintaining relationships between source and copied content. Nevertheless, customers and integrators have reported serious and ongoing problems with this feature, especially when translation and globalization are involved. You’ll want to test carefully. If you are looking for sophisticated multilingual/multisite management, you might want to compare Percussion against SDL or Adobe.

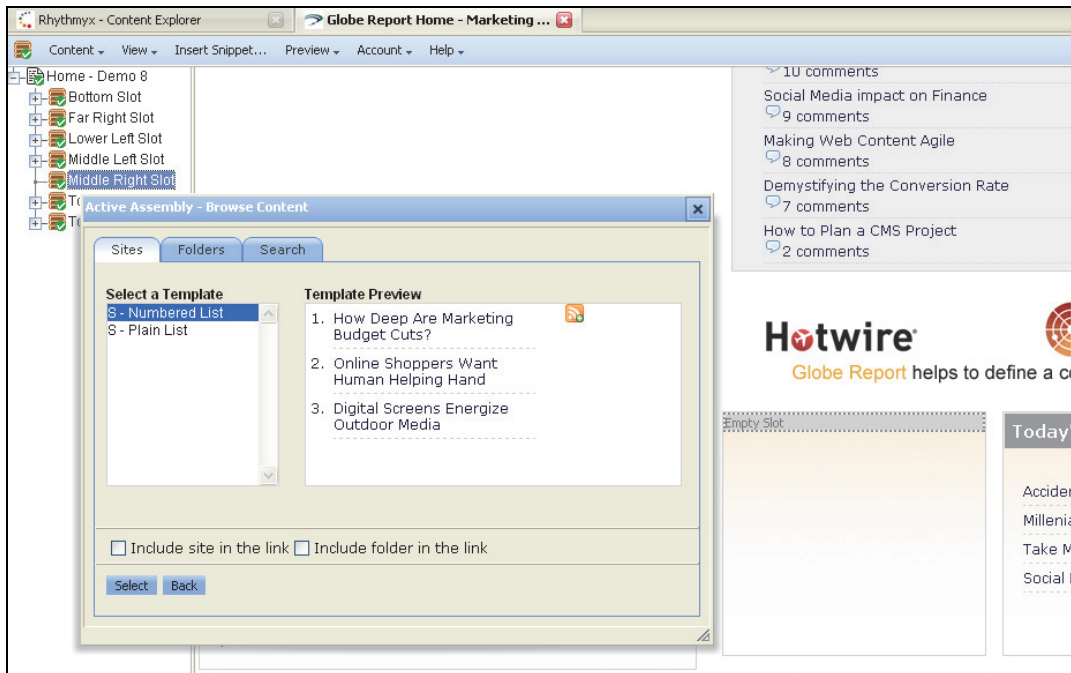


Figure 134. The Active Assembly interface (see text for discussion.).

For search, Percussion employs a customized Lucene solution (Figure 135). The point-and-click UI for query building is reasonably powerful, and allows searches to be constrained by date ranges in addition to many standard sorts of metadata fields. Searches can be saved, and the user never sees or has to deal with raw query expressions. Since the search subsystem is brand new, however, you should plan to do some careful testing of it.

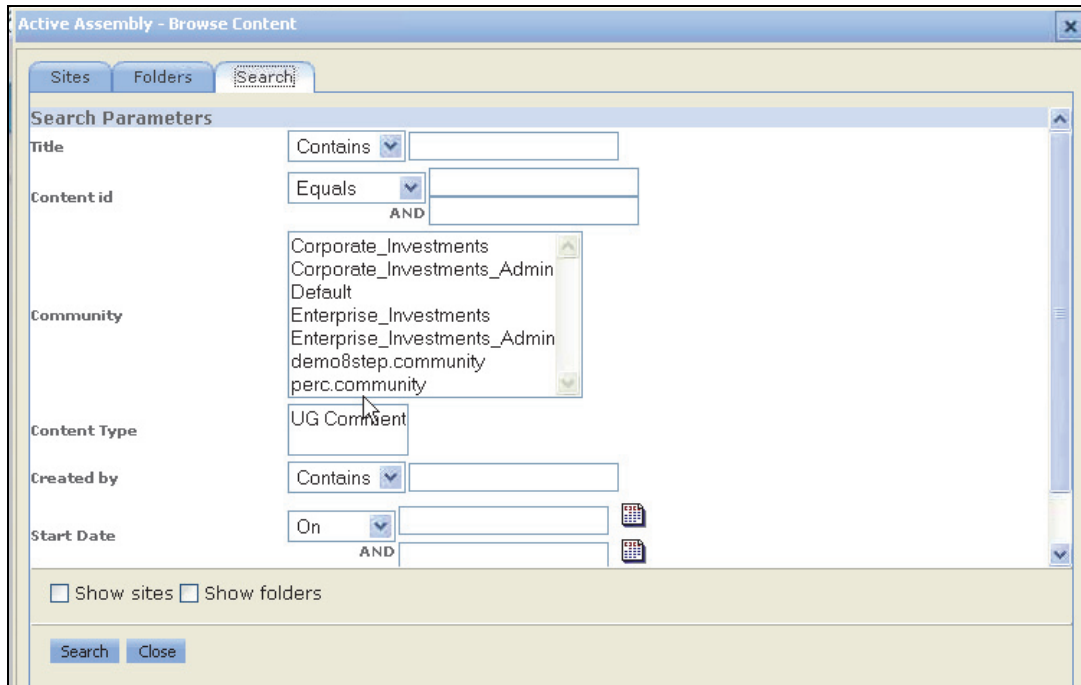


Figure 135. Content search is now Lucene based and supports searches constrained by date range, among other criteria.

Authorized users can skip workflow entirely via a “Publish Now” feature. From the in-context editing environment, you can modify (edit) an individual content item, then push the item in question out to the delivery server at the press of a button.

The item goes live immediately, without further review. Obviously, this gives appropriately permissioned individuals great latitude in how and when content is published. If someone finds a typo in a content item, it can be corrected immediately, without the item being workflowed all over again. Likewise, if analytics are showing that a certain content area is hot, you can modify one or more pages right away to take advantage of that fact. However, the speed and flexibility come at the cost of governance concerns and more complicated event tracking. Audit trails for content items (showing who edited or published them, and when) are always available in the Explorer interface — but there will be a lot more of them if contributors are constantly making tweaks in near real time. Of potentially greater concern, though, is the fact that content that is pushed out via “Publish Now” isn’t staged or reviewed, and it bypasses normal processes. Therefore, your “normal processes” have to be liberal enough (from a governance standpoint) to accommodate the “Publish Now” metaphor. Otherwise, the feature isn’t really a feature, and will need to be disabled.

If your organization can accommodate Publish Now, though, it can be quite powerful when used in conjunction with the system’s new Linkback feature. If you are viewing a live web page in the browser pane of the Active Assembly interface (or within Percussion’s Web Analytics interface), a click on the “Linkback” button will take you immediately to the same page in the *production* environment, where edits can be made.



Figure 136. The Publish Now command does just what it says; it Approves content, marks it Public, and immediately sends it to delivery.

Experience

Publishing

The system’s good support for content reuse and XML content transformation make it comparatively well suited to multichannel publishing.

Percussion’s Community Marketing Solution (sold as an add-on) captures user ratings and recommendations, and facilitates the creation and management of RSS/Atom feeds. The system can be set up to import content directly from an RSS feed. Conversely, it’s easy to expose CM System content via feeds.

Digital Marketing

As previously mentioned, Percussion sells an optional personalization module. When personalization rules are applied at various steps during assembly, some powerful page tailoring can happen at delivery time, where every subcomponent of a rendered page — and every subcomponent, can be custom tailored to a repeat visitor’s profile.

Obviously, this kind of thing can be taken to an extreme: You could quickly get to the point where no two site visitors ever see exactly the same page content when visiting the same URL. There’s also a price to pay in performance for excessive dynamism, and depending on how many page components there are and how they’re bound, caching can be very tricky. Generally, however, most Assembly rules are executed at “parbake” time, and Percussion claims that granular items are preferentially pulled from the cache at page-request time, tending to mitigate performance issues. Still, if you intend to serve highly dynamic content,

Experience	
Publishing	
Standards Adherence	<input type="checkbox"/>
Multichannel	<input type="checkbox"/>
Mobile	<input type="checkbox"/>
Digital Marketing	
Site & Campaign Analytics	<input type="checkbox"/>
Testing & Optimization	<input type="checkbox"/>
Segmentation & Personalization	<input type="checkbox"/>
Social Media Integration	<input type="checkbox"/>
Promotional Campaigns	<input type="checkbox"/>
Community & UGC	<input type="checkbox"/>
Workplace	
Collaboration & Networking	<input type="checkbox"/>
Dashboard	<input type="checkbox"/>
Ancillary	
Site Search	<input type="checkbox"/>
Online Forms	<input type="checkbox"/>
Module Ecosystem	<input type="checkbox"/>

you'll want to do some careful testing to get an idea of what the system's performance characteristics are.

In a highly dynamic, persona-driven system of this kind, it's important to be able to preview pages as they might appear in various content-item permutations. This brings up another issue: How *do* you test that many combinations of page presentation? There is no easy answer, unless you add the new "Percussion Personalization Solution," which introduces a new "dynamic preview" user interface to solve this problem. It enables users to specify visitor profiles in a left side pane, or click through the site to build up a profile on the fly, and then see the corresponding dynamic pages the visitor would see on the site. This tool enables the business user to test how the visitor profile is *updated*, based on how a visitor interacts with the site.

Without the Percussion Personalization Solution, you're left with taking more of a divide-and-conquer approach, previewing individual content items on their own before combining them into more complex, compositional views. To Percussion's credit, authors can preview content items in all of the different renditions where they will appear, and any edits that are made to an item will cascade throughout the system as expected.

Percussion is very much attuned to the fact that web-based marketing has lately become much more about feeds, real-time interaction with visitors, and user-generated content (even if it just involves preferences, tags, ratings, or recommendations), as opposed to the more old-fashioned "push" style marketing. (This may partially explain why the CM System solution set is generally lacking in push-style tools for email campaigns, e-newsletters, and various opt-in programs.)

Percussion now has a set of pre-packaged black-box delivery tier functions called Online Interaction Services. These are optional and purchased separately through one or more Solutions. Online Interaction Services (which play in a loosely coupled Web 2.0 "mashup" style with your existing or new Web delivery application code) include dynamic content matching based on visitor profiles, visitor behavior tracking (creation and update of visitor profiles, based on visitor behavior), analytics data capture, user-generated content, and RSS feed importer services between the Web Delivery and CM tier. Online Interaction Services can be invoked by dynamic web application code at the server side, or they can process client-side through DHTML tags embedded in pages, much like Ad Servers.

Percussion offers a nicely integrated analytics solution through a partnership with Webtrends. From within the Webtrends SmartView UI (which allows you to see traffic info for individual elements on a page), the user can click directly into the Active Assembly in-context authoring interface to modify the page or elements within it. The combination of analytics insight and fast content modification gives marketers short iteration times and allows for quick optimization of pages. The short iteration time is one of Percussion's differentiators.

Percussion now offers a Community Marketing Solution that incorporates blogging, including publishing into any Blog using the MetaWebLog or Atom API, as well as harvesting blog comments into the CMS, RSS generation, and bulk External Content Capture, which imports content on a scheduled basis from RSS feeds. This doesn't mean you automatically gain the benefit of every CM System feature. If you want to manage tags or metadata associated with blog entries, workflow, or apply lifecycle logic, you may find that you need to do extra work.

Ancillary Services

For site search, you need to employ your engine of choice.

There is no third-party module ecosystem, which is perhaps indicative of the niche-y nature of this tool.

Vendor Intangibles

At about 60 employees, Percussion is smaller than other competitors in this tier are. However, Percussion can surpass some peers in both geographic reach as well as longevity. The company reports that its business is split 60:40 between North America and EMEA.

While Percussion has begun to support user groups, including a few beyond North America, some customers have found support — especially online support — a bit thin. With a smaller base, irregularities in the products sometimes take a bit longer to surface and be resolved. Also, there appears to be a scarcity of Percussion-trained integrators, and customers seem to rely on Percussion’s own professional services arm heavily. The company says it’s focusing effort on beefing up its user community.

In the past, Percussion has avoided building relationships with systems integrators and other partners, and this tendency has earned them a reputation — deserved or not — of having a closed and quirky culture. The company now is building relationships slowly with systems integrators. Still, prospective North American customers, in particular, will find comparatively few systems integrators who can claim expertise and a close relationship with Percussion. Customers typically tend to buy deep expertise from Percussion professional services, and then build project teams from in-house staff or individual contractors. If your enterprise is developer thin, you will want to explore your alternatives carefully here.

Typically, with every enterprise deal, Percussion will include a quickstart professional services and a training package that includes a three or four-day developer class, as well as some help from a Percussion architect. Failure to take advantage of this would be foolhardy. While the platform is straight Java, it’s complicated enough that you’ll want to have ready access to practitioners with genuine Percussion experience.

With respect to overall strategy, Percussion still seems to be struggling mightily to find a way to “break away from the pack.” CM System’s “secret sauce” is hard to convey properly in a 30-second elevator pitch. Recognizing the need for some new thinking, Percussion underwent a realignment of top management in 2008–2009, and again in early 2012. Moreover, supporting two products will almost certainly stretch the company’s resources.

Licensing runs from US\$72K for smaller sites to \$150K (Enterprise). However, note that there are extra fees (\$30K to \$50K per Solution), should you wish to make use of some of the optional Solutions modules.

Percussion is perpetually rumored to be up for sale, but it is closely held, does not appear to be actively shopping, and it could probably chug along with the current ownership for quite a long time. However, since financials are not public information, there’s really no way to know

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

whether Percussion might be suffering the same license-revenue erosion that some other Web CMS vendors are experiencing in the current market. The company claims to be successful so far, but you'll want to investigate.

Conclusion

As a Java-based, decoupled system, Percussion CM System is somewhat similar to HP's offering — except that Percussion:

- Is less expensive (and doesn't require client licensing)
- Uses a database to store your content
- All content (at all granularities) can be repurposed in other formats, using Velocity templates
- Has a cleaner, more modern-feeling architecture

Percussion has also gained some insight by observing HP, particularly in the way that it now accommodates scenarios with heavily unstructured content. The company has replaced its previous "Quickstart" methodology — which featured potentially complicated content modeling and XML migration — with a reference implementation called "FastForward," designed to get customers started quickly with a basic set of content types that they can modify and add on to later. For some users, this might be an attractive way to skip a certain amount of time-consuming, content-modeling work up front, and thus go from a standing start to a testable system in reduced time.

However, the real power in the system — and your consequent return on investment — comes when you have done the work necessary to chunk your structured content and "slot" it for meaningful reuse, and that can take time (especially if you decide to exploit the extra personalization and other features that come with the add-on modules).

Many prospective customers will welcome Percussion's new emphasis on digital marketing and interaction. You should just remember that a "platform" approach is deeply bred into the DNA of this company, and we suspect Percussion will continue to make choices that appeal much more to the developer (and super user) mind-set than to the everyday editor or manager. That's not necessarily a bad thing, but you should be prepared to evaluate the product in this light. Similarly, exercise some caution with the new CM1 product, particularly since it doesn't seem to maximize on Percussion's core (engineering) strengths.

If you are looking for a system to power a collaborative or document-heavy intranet, Percussion is not a good fit. On the other hand, if personalization and true content reuse are hot-button issues, CM System is worth investigating, particularly in the context of Multichannel Publishing and Ultra-Large Single Site scenarios, where it has the potential to excel in the hands of an enterprise with deep Java and XML skills.

Just make sure that you have your own project management ducks in a row. You're likely to find Percussion CM the product more impressive than Percussion the company, which has a history of being comparatively quite erratic.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

SDL: SDL Tridion

www.sdl.com

Vendor at a Glance

Specsheet	SDL: SDL Tridion 2013 Summary
Geography	Global, with emphasis on Europe & North America
What's New	<ul style="list-style-type: none"> • Company released new edition, with somewhat refreshed UIs tailored to user groups, and advanced workflow services
Strengths	<ul style="list-style-type: none"> • Comprehensive support for XML and “Schema embedding” can facilitate comparatively good content reuse and re-purposing • Flexible workflow system supports complex approval processes • Well suited to multinational firms with substantial globalization needs and/or mixed Java/.NET delivery tier environments • Loosely coupled architecture may suit some enterprise infrastructures • Blueprinting functionality provides very good support for multisite management in highly distributed environments • Unusually good support for full-cycle email newsletter publishing • Developer extranet and regional user groups are well regarded
Weaknesses	<ul style="list-style-type: none"> • Complete solution may require multiple additional licenses above the introductory package; be careful of sandbagged initial bids • Entitlements subsystem is surprisingly underpowered for the kind of global, distributed operations the company targets • Kaleidoscope of different technologies (COM, Java, .NET) under the hood adds substantial complexity to development and maintenance • Experienced Tridion specialists are globally in relatively short supply • User interface is comparatively complex and difficult to use, despite recent redesigns and refreshes • Company emphasizes functional development over (long-postponed) underlying architectural improvements • Consistently lags behind major competitors in digital marketing
Potential Fit	Multichannel Publishing, Global Enterprise
Unlikely Fit	Simpler and Mid-Range Scenarios, most Digital Workplace Scenarios
Compare To	HP, Adobe, Oracle, Sitecore, OpenText
Operating Systems	Windows, Solaris, IBM-AIX, HP-UX, and Red Hat Linux
Repository	Database: Oracle, SQL Server, IBM DB2
Client	IE, Firefox, Chrome, and Safari
App Platform	COM & .NET in management tier, Java/.NET in delivery tier
Licensing	Server, CPU, user, and module based, with typical deals starting at \$150,000–250,000+
Ownership	Public (LSE: SDL)

Summary

Tridion is one of the oldest players in the upper tier of the web content and experience management marketplace, along with (HP’s) TeamSite and (Oracle’s) FatWire (now rolled into WebCenter). Of course, all of them have been acquired. The Dutch-rooted Tridion was acquired in 2007 by the UK-based SDL plc (where it was transformed into the “Web Content Management Division,” and later merged with SDL Structured Content, Fredhopper, and Calamares Media Manager to form the Content Technologies Division). SDL was a well-known Translation Management Systems (TMS) vendor and service provider that has since diversified its portfolio.

The SDL Tridion WCM system is oriented explicitly toward digital managers in global enterprises. When its numerous optional modules are taken into account, the product is as feature rich (and complex) as virtually any product in this Report. Just remember that like its peers, Tridion is not a turnkey solution. It will require substantial time and effort to set up. However, if you can accept a sizable up-front implementation burden and the commitment of development resources on an ongoing basis, SDL Tridion can be used to build impressive websites.

SDL Tridion is best suited to Global Enterprise Scenarios, where workflows that culminate in simultaneous publishing of content in multiple languages might be de rigueur. It is almost certainly overkill for informational or single-domain/single-language corporate sites, and less suited for advanced digital marketing scenarios as well. Moreover, a lack of social and collaboration services makes SDL Tridion increasingly ill fitted to most Digital Workplace Scenarios.

Scenario Fits	
Simpler Site	
Informational	<input type="radio"/>
Microsites & Landing	<input checked="" type="radio"/>
Mid-Range	
Basic Digital Marketing	<input type="radio"/>
Mobile Site	<input checked="" type="radio"/>
Community Oriented	<input type="radio"/>
Complex Site	
Advanced Marketing Portal	<input checked="" type="radio"/>
Global Enterprise	<input checked="" type="radio"/>
Multichannel Publishing	<input checked="" type="radio"/>
Ultra-Large Single	<input checked="" type="radio"/>
Digital Workplace	
SMB/Departmental	<input type="radio"/>
Global/Enterprise	<input type="radio"/>

Introduction

Founded in 1999, in Amsterdam, Netherlands, Tridion spent a number of years building a solid customer list in Europe before venturing out into the North American market in 2006. By 2011, the majority of its business started coming from North America, with Europe becoming less important in terms of sales in general and new customers in particular. To support its North American growth, the company moved some of the top talent from R&D and Professional Services into its U.S. offices. Despite all of the effort to build up internal expertise and external partner channel, the availability of skilled resources in the region remains limited.

SDL sells a host of product variants under the SDL Tridion brand, including some that can be traced back to previous consulting work. Figuring out just which modules you need can take the kind of effort typically reserved for enterprise-level packages. Various optional modules help you manage print publications, email campaigns, and e-commerce, among others. Nevertheless, these offerings all revolve around the main web content management system.

Like MS SharePoint, Tridion tends to name by its release year. SDL released the 2011 version in January of that year, and it was a substantial and rather overdue upgrade to the product’s user interface — but not to its glued-together and rapidly aging underlying architecture. In 2012, SDL skipped a year and didn’t release the expected 2012 version, in favor of only a few features bundled in a service pack. In 2013, the next official release came out bringing changes to workflow, further enhancements to role-tailored UIs, and support for the CMIS standard.

Technology

Technical Administration & Security

SDL Tridion is fundamentally a decoupled system, with management, production, and staging occurring behind a firewall, on dedicated hardware that is separate from a delivery tier that essentially consists of a series of Tridion-specific APIs.

This brings pros and cons. On the one hand, you can put Tridion comfortably behind diverse front-end environments without upsetting existing apple carts. It also allows different digital properties in your enterprise to exhibit various levels of sophistication — an important consideration for the heterogeneous, global enterprises that SDL targets. On the other hand, this separation makes it harder for Tridion to do fancier stuff on the visitor-facing side, and puts some hurdles in the way of digital marketers who want to control user experiences from a WCM dashboard. Read on for more details...

Once again, the product is really two distinct environments. On the content management side, the core technology is all Microsoft — a mix of much dated COM pieces and, increasingly, more modern .NET components. As long as SDL Tridion chooses to support VBScript templates, you will not escape the COM dependency. The vendor doesn’t recommend using VBScript in newer implementations, and C# and VB.NET are the preferred options. In fact, the majority of legacy implementations are struggling to move away from VBScript to more modern scripting technologies.

On the delivery side, you can choose between Microsoft (.NET) or Java. Intriguingly, some licensees report that the company gives more love to the Java side. In any case, the presentation layer is fully decoupled from the management layer; once content is deployed, there is no back-traffic across the firewall, with the possible exception of a specialized forms module. The presentation server itself runs on top of standard application servers, including IBM WebSphere, Oracle WebLogic, and Microsoft Internet Information Server (IIS). This obviously offers a certain amount of flexibility for multinational conglomerates (and others) with heterogeneous server environments, and some customers do run Tridion on both Java and .NET delivery environments.

With respect to publishing, content deployed to SDL Tridion’s Presentation Server is stored in either a file system or a relational database (Oracle, Microsoft SQL Server, or IBM DB2).

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

The process of grabbing the XML content from the repository and applying the various (page and component) templates before pushing it out to the delivery tier can be a processor-intensive task, performed by the “Publisher” service. Since the publisher task runs through its queue of publish events sequentially (each event triggered by users pushing the “publish” button), templates that take long to execute may delay other content from getting online quick enough. The milliseconds of each separate component on a page quickly add up, especially in an implementation heavy on VBScript, and republishing a large site may take hours — or even days. For this reason, large multinational companies often run several separate publishing servers — with attendant increases in licensing costs.

Access Control and Security

SDL Tridion’s access controls can handle the basics (including LDAP integration), but oddly, the product has no notion of roles per se, just groups. You can set up user groups that mimic roles (e.g., author, developer, admin) across the entire installation as a work-around. To be sure, Tridion groups are extensible, but the non-intuitive way that you address roles will make you spend hours designing your security model — and more hours implementing it — and still more hours educating system participants about the particularities of their permissions. Note for example, that the special role of “administrator” will be a super user across all of your sites, not just one.

Mapping LDAP groups to SDL Tridion rights isn’t very flexible (you may find it easier to change the structure of your directory service). This is a frequent limitation, however, and it hasn’t prevented some of the largest multinational companies that use the product from achieving complex publishing goals.

System reporting is relatively basic in the UI, and mostly revolves around items’ versioning history and the list of items published by the “Publisher.” Since this contains all data on publishing actions (including the user and date/time), filters allow you to have a relatively good insight into what happens in your environment.

Development

On the whole, SDL Tridion has good APIs for consuming and exporting repository content, thereby making XML-based integration with (for example) ERP systems easier than in some competing systems. While SDL Tridion has never supported Java standards like JCR (Java Content Repository), as of the 2013 release, they now support CMIS for external content libraries. What this means in practical terms is that you may be able to access content in other repositories from Tridion. However, because CMIS is mostly a document management-oriented standard, its applicability to WCM and SDL Tridion is rather limited.

In version 2011, SDL introduced support for the new Open Data (“OData”) protocol in an attempt to open the proprietary system and its APIs for sharing data across multiple repositories. For example, in the current state you can use it on the content delivery side to call a Web Service to pull content from third-party content providers into portlets on an SDL Tridion page. However, the biggest drawback here is that you must install a Java application server to use it (JBoss or Tomcat). This may not be a problem if you are already a Java shop, but might be hard to justify in a Microsoft environment, and run this Java-only implementation in a Java container as opposed to a more comfortable IIS approach. Not surprisingly, there’s not much uptick with OData so far. As one developer puts it, “I just refuse to touch it.”

Although the system as a whole is based on XML, SDL Tridion has had some problems dealing with complicated, heavily nested composite documents. If you work with compound technical documents a lot, you'll want to see if SDL Tridion will allow you to decompose them (and reconstitute them again) properly before facing a choice of buying extra modules such as SDL's own Trisoft DITA or LiveContent products, or committing to this system.

To be sure, SDL Tridion's use of XML technologies effectively supports most component-based web publishing scenarios, and this remains a strength of the platform relative to other competitors, mostly notably Sitecore. However, don't consider Tridion a full compound document management system for managing complex engineering (or other) highly structured documents.

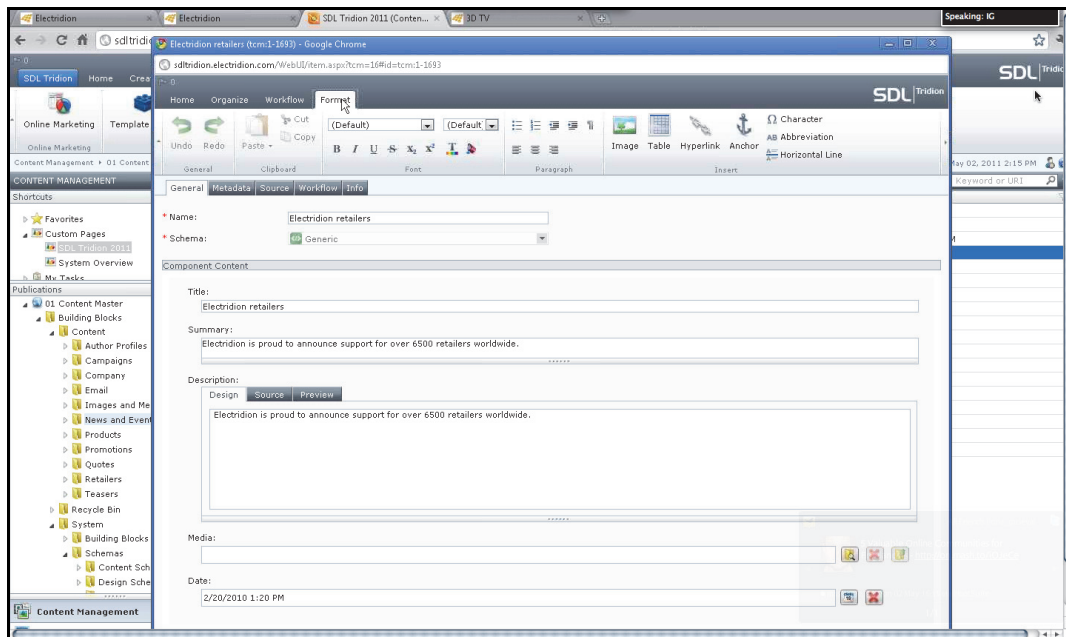


Figure 137. Editing a component based on a certain schema — a combination of efforts from developers and designers to be used by content authors.

Customers who need to manage multiple and/or multilingual sites from a central repository have praised the product's "BluePrinting" approach for its ability to allow the creation of new country, language, brand sites, or microsites based on specific parent-child relationships and rules of inheritance. Those that do their homework thoroughly can use the combination of modular templating and BluePrinting to update the branding of dozens of sites. Licensees with single sites and simpler requirements, on the other hand, have often found that BluePrinting is overkill.

Just understand that BluePrinting can become a rather tedious and complex process, especially for larger multi-site or multilingual installations. You must be sure you get it right the first time, as it is not a trivial task to change the BluePrint later. Overall, this process isn't always as magical as the snazzy interface suggests. Indeed, SDL salespeople frequently show the Visio-like Blueprint visualization interface, but in the real world, you won't use that. Instead, you'll use a comparatively quite powerful, forms-based interface to manage variable inheritance.

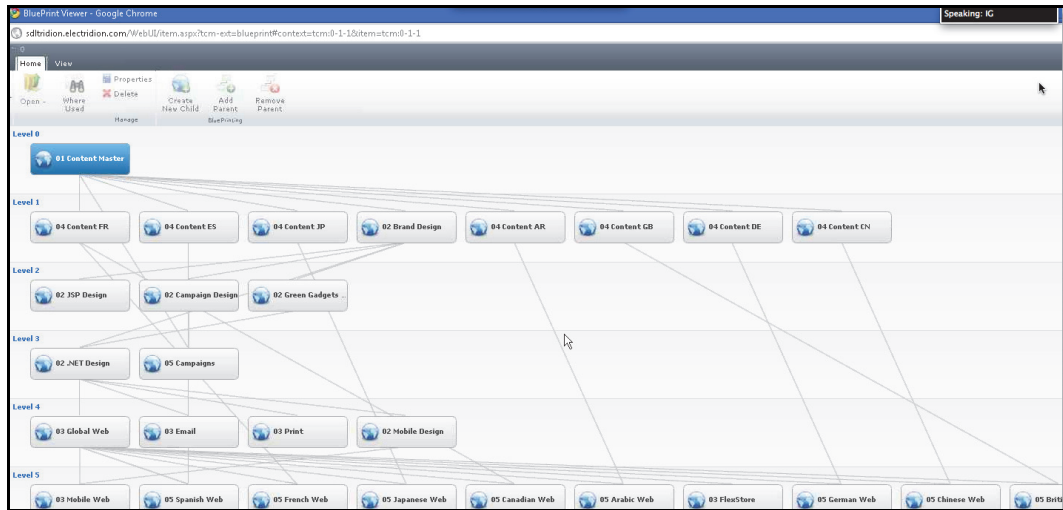


Figure 138. A typical BluePrint example from the SDL Tridion WCMS demo.

BluePrinting is coarse grained in the sense that publications inherit from ancestors in all-or-nothing fashion. This means that in a real-world implementation, the BluePrint design can be very busy. The implications of the BluePrint design can be hard to fathom. Since there is no other way than this inheritance to share content between publications (you can't even copy/paste an item from one publication to another), complex reuse and localization scenarios will require some very creative thinking. Several large projects in the past have had to be stopped and redone from scratch after the sudden realization that the BluePrint wasn't quite working out. If you bought this product for its vaunted multi-site management capabilities, you'll definitely want to invest in BluePrint training, as well as Professional Services.

Content types in SDL Tridion are represented as XML Schemas. A schema is a set of fields that produce appropriate content types like articles or press releases. Schemas need to be designed by developers at the beginning of your implementation, but they can be difficult to modify later, with variable impact on existing content items. Plan carefully. You can also use an external XML editor and register an externally created XSD, but that mainly applies to schemas that are more complex.

Be wary of large unstructured blobs of rich text content for authors to fill as they see fit. While the system can support this approach, it was not designed for these types of unstructured content, which will create more unnecessary work for your content contributors. SDL Tridion will tend to favor those customers that enforce componentization and strictly structured separation of content and design.

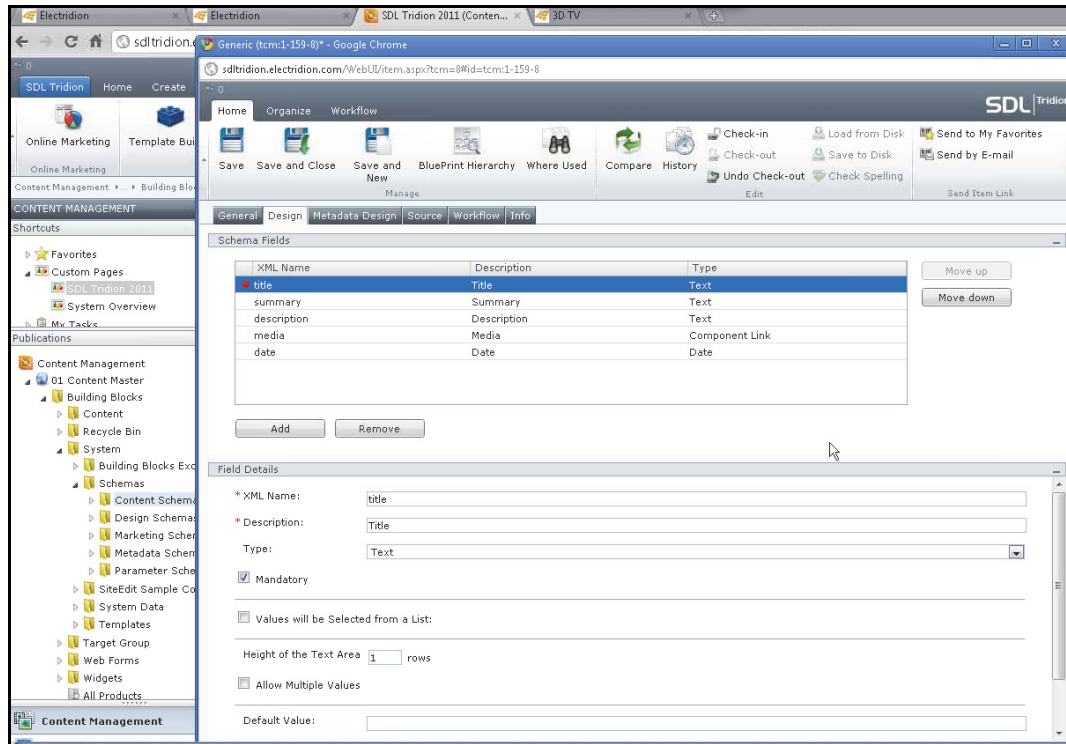


Figure 139. Building content types in a browser, including controlling element behavior.

On the templating side, the scripting languages for templating and customization have historically included a mix of VBScript, C#, and XSLT. In addition, you can use Dreamweaver templating and JScript.

The company advocates the use of modular templating. This gives developers the ability to compose a template out of reusable template layout and code snippets. The primary benefit of modular templating is that it allows designers and developers to do what each does best, using different tools. It could also enforce a clean separation of presentation and logic.

The risk is that this effectively turns site design into an exercise in large-scale programming. (CoreMedia, OpenText, and Oracle have similar shortcomings here, but perhaps not as intense.) If you were already planning a major development effort, modular templating could save you time. It could also be worth evaluating in cases where you expect to redesign a site frequently. However, we suspect that for customers who are more interested in content reuse than template/logic reuse, it will only introduce complexity into a process that's already complex.

Unfortunately, there is no real choice; you will be forced to use at least .NET (for certain functionality) and XSLT (for performance) for both the page templates and the component templates. The role of .NET and XSLT template building blocks is to generate HTML, JSP or ASPX/ASCX fragments from the XML content for the presentation server to display. This complexity creates a matryoshka doll of nested code. Developers complain that working together with the different fields of expertise that is needed to create an actual page poses a very unwelcome challenge.

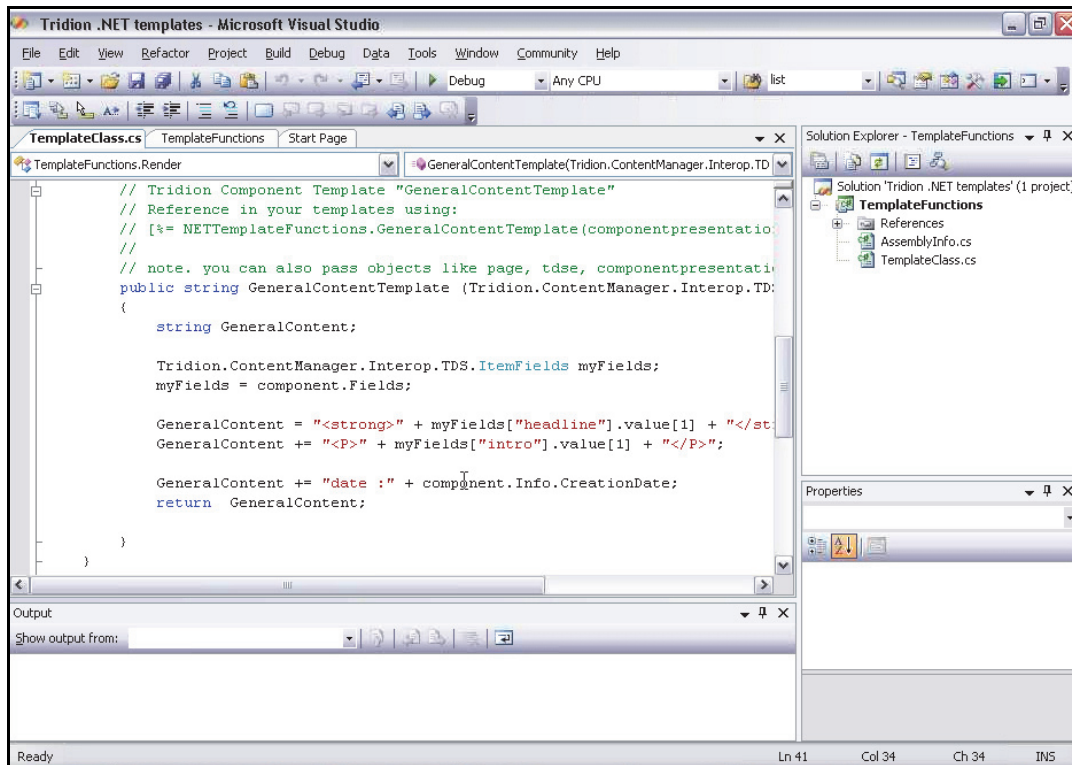


Figure 140. Editing a C# template that uses .NET controls in Visual Studio.

In part to mitigate these problems, SDL Tridion has introduced an extra design and debug tool, the Compound Template Designer. The “template building blocks” can be mixed and matched to create a “Compound Page Template” in Tridion’s Compound Template Designer interface, which will time the performance of the separate elements. By throwing this extra tool into the mix, the vendor hopes to stimulate the kind of structured and planned development the product really requires. To the extent that some developers ignore the methodology SDL preaches in its training, the Template Designer could help enforce the best practices — but it also presents yet another tool to understand and master.

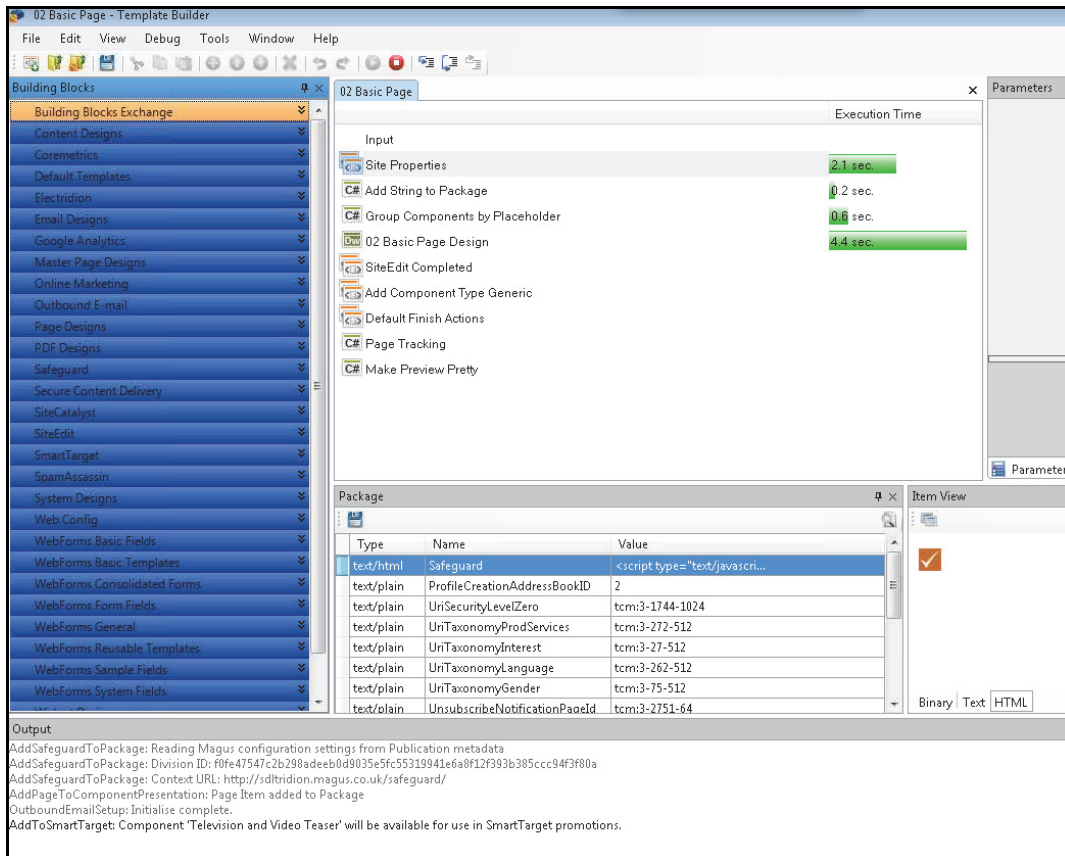


Figure 141. Final templates can be composed from reusable “building blocks” (XSLT, scripts, .NET components, CSS, etc.) in the Compound Template Designer interface.

As with any type of content in the system, “template building blocks” will have versioning (or workflow), which some find to be a worthwhile advantage.

Performance

Client-side performance has traditionally been a problem with Tridion. It still is. The platform’s back-end consists of an ASPX codebase running on IIS, as of the 2009 release. With the 2011 release, the interface largely consists of client-side JavaScript (communicating with the server through WCF).

As a Windows Server-based platform, you’ll find you can tweak SDL Tridion server performance and behavior extensively using standard Windows tools, including dividing services on different machines to create more distributed architectures. SDL Tridion comes with a Microsoft Management Console plugin, which enables remote administration, as well as a familiar interface for those used to working with Microsoft Management Consoles (Figure 142).

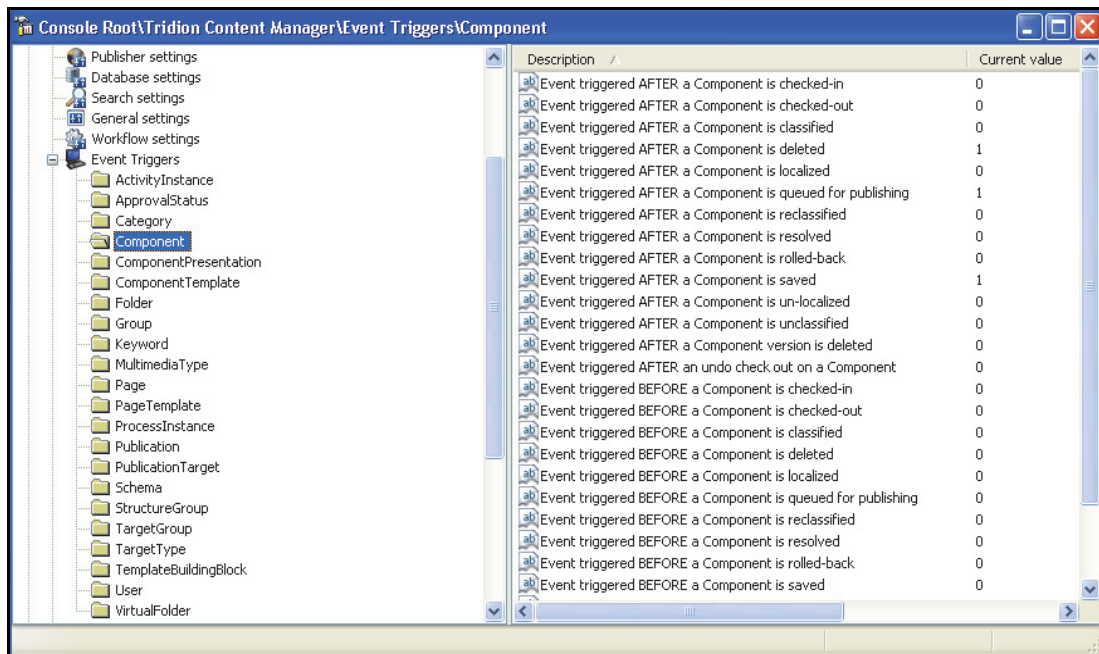


Figure 142. Microsoft Management Console.

Now let's turn to the delivery side. SDL Tridion uses an interesting cache architecture based on Java Messaging Service. In this (optional) setup, Presentation Server instances *subscribe* to updates posted to a JMS queue. As newly updated content lands on the queue, server instances pull it off as needed; otherwise, cached content is served by the local machine from its own local cache. IT experts who are familiar with the quality-of-service benefits of messaging oriented middleware will appreciate the reliability implications of this architecture.

This strikes us as a sound approach — one well worth looking into if you're building complex, mission-critical sites that have to perform well under a heavy load. Just note that it's a Java-only solution, aimed at customers who are running clustered Java app servers.

Content

Contributor Experience

The latest major UI refresh took place in the 2011 version, when the MS Office-like ribbon graced the UI. The graphite-colored (yet customizable) UI has contextualized buttons and actions, as well as status messages — but not many worthwhile changes aside from that. In short, the SDL Tridion UI remains one of the most difficult to use among the upper-range WCXM offerings that we cover in this report.

In 2013, SDL started making changes to its editorial flow. The primary complaint most editors had with Tridion was that the highly componentized system makes them jump a plethora of hoops to get anything published. For example, creating a simple news page could involve as many as twelve tasks with dozens of clicks. Being a component-based (vs. a page-based) system, SDL Tridion is click-intensive. The vendor started responding to this complaint by hiding some of its component-based complexity behind the scenes and presenting content in a *faux* page-based view.

Content	
Contributor Experience	
Overall Usability	●
UI Accessibility	●
Contributing Content	
Authoring & Transformation	●
Tagging & Taxonomy	●
Content Reuse	●
Media & Document Management	●
Repository Services	●
Content Lifecycle	
Workflow	●
Globalization	●
Archiving & Compliance	●

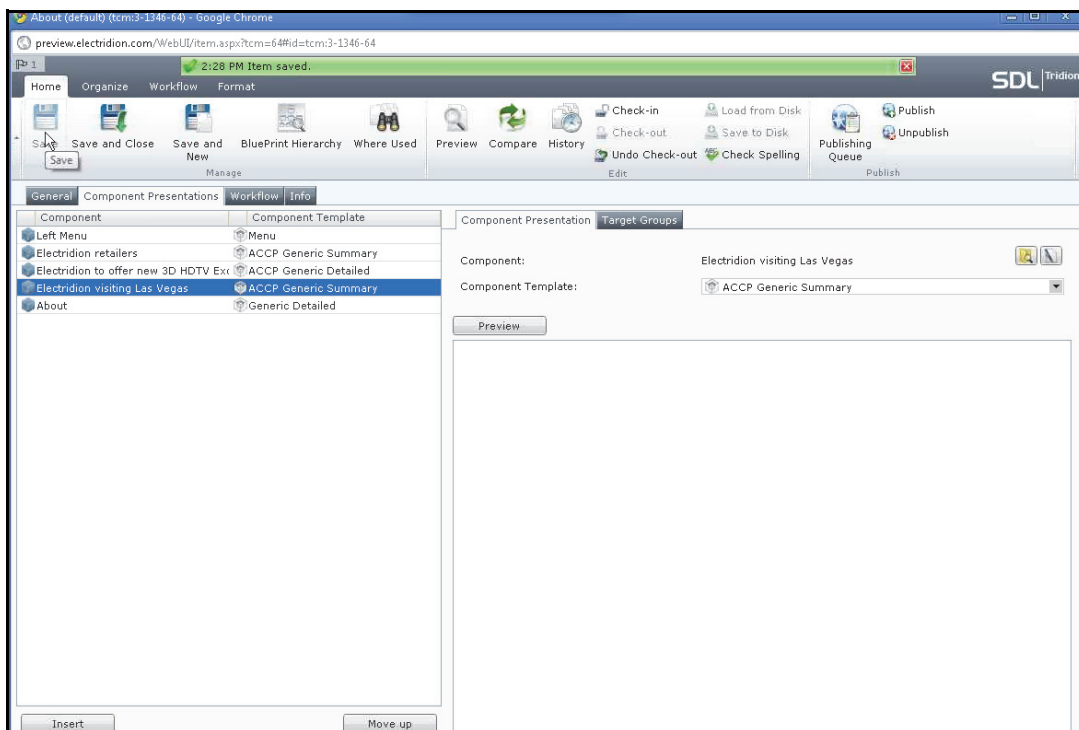


Figure 143. Let your content editors test the interface to see if they appreciate bright orange and green status notifications that were added in the 2011 release.

SDL Tridion introduced the notion of group-tailored interfaces, in the same vein of trying to hide away the product's complexity from casual contributors and digital marketers that didn't feel comfortable using the admin interface.

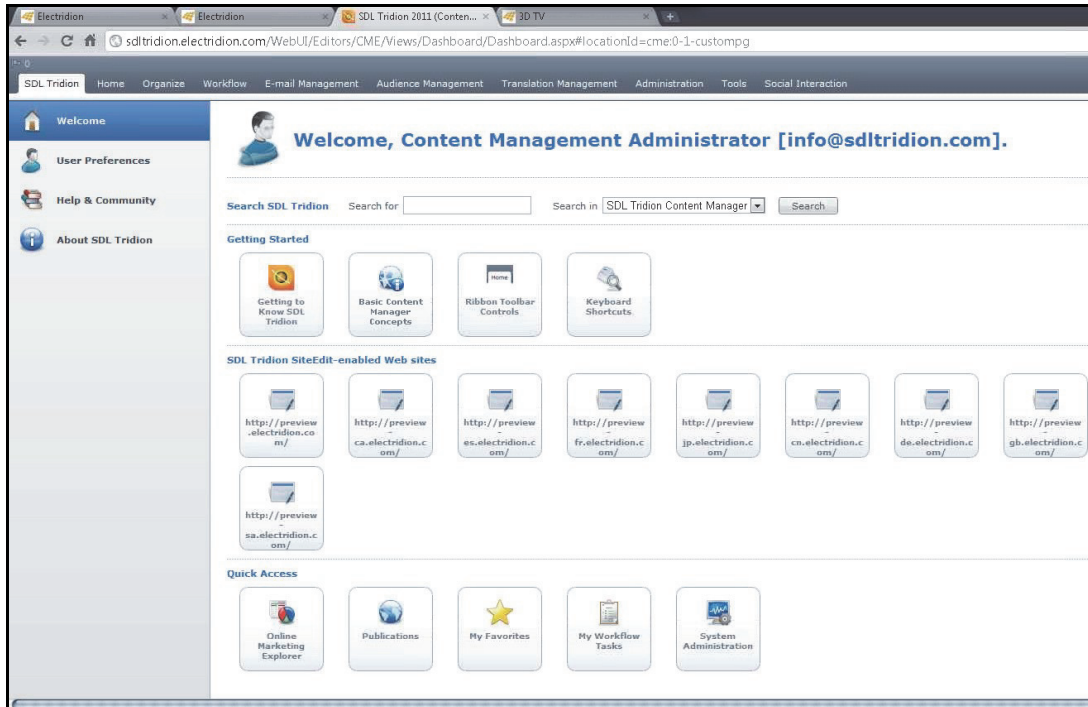


Figure 144. Mimicking the Adobe Experience Manager, SDL Tridion's added homescreen is tailored per user role. This is an admin's starting point after login.

Therefore, don't expect to run it on a low-spec client PC — you will need plenty of memory, and users often have to restart IE in order to get the system working again.

The client also needs a lot of bandwidth to be responsive; recent versions have taken steps to improve this, but there's a lot of XML communicated between client and server. In an office environment, this may seem trivial, but for field offices on low-speed connections, it can turn a simple task into a tedious test of patience.

SDL Tridion's interfaces are powerful, but they tend to be complex (with many tabbed panes packed inside a single iframe, for example), and you often will find yourself going many levels deep into nested dialogs to do a certain task. In addition, it's almost impossible to do anything in the admin interface without seeing annoying pop-ups with alerts and confirmation boxes, and Internet Explorer struggles to keep up with all of these dialogs and windows.

The client also needs a lot of bandwidth to be responsive; recent versions have taken steps to improve this, but there's still a lot of XML communicated between client and server. In an office environment, this may seem trivial, but for field offices on low-speed connections, it can turn a simple task into a tedious test of patience.

Therefore, the browser-based admin interface (Figure 146) remains comparatively rich, although certainly not without shortcomings. Befriend those pop-ups and modal dialogs; you will see plenty of them.

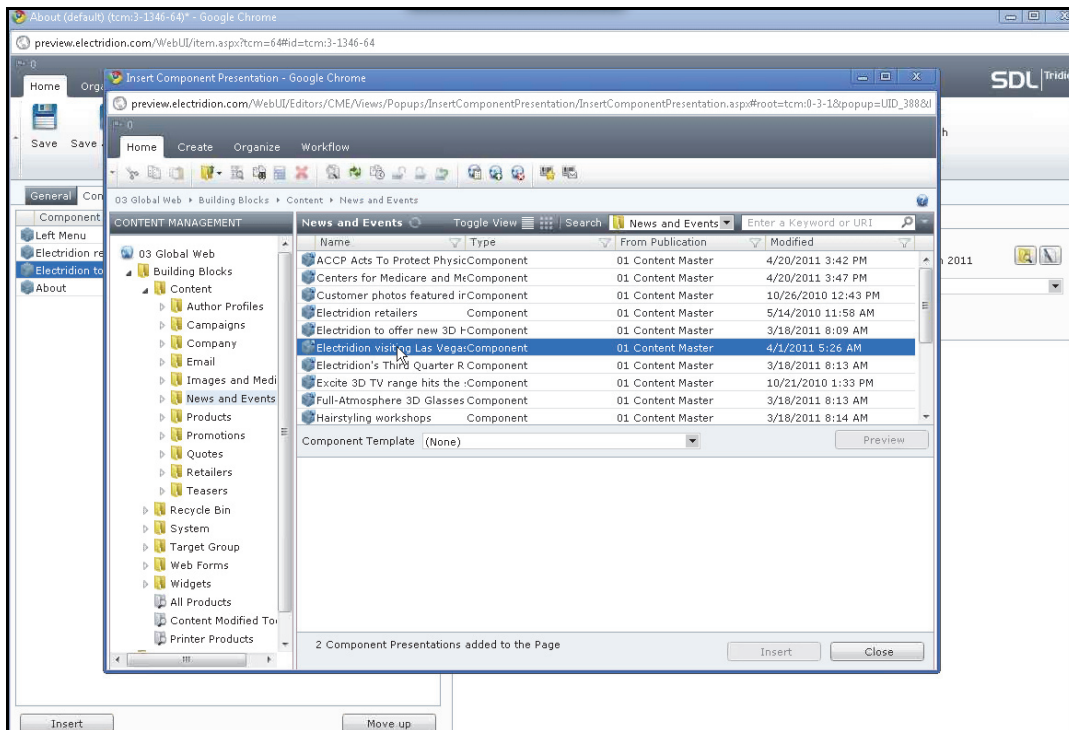


Figure 146. Content editors are not always happy with SDL Tridion's clicking and pop-up intensive experience.

Therefore, don't expect to be able to run it on a low-spec client PC — you will need plenty of memory, and users often to have to restart IE in order to get the system working again.

The client also needs a lot of bandwidth to be responsive; recent versions have taken steps to improve this, but there's still a lot of XML communicated between client and server. In an office environment, this may seem trivial, but for field offices on low-speed connections, it can turn a simple task into a tedious test of patience.

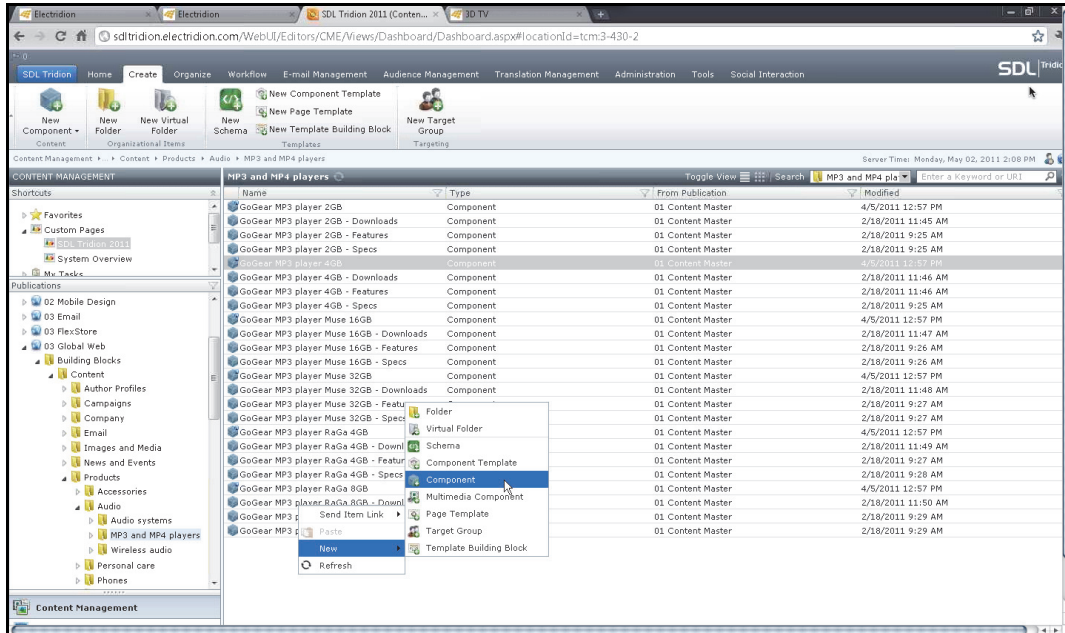


Figure 147. SDL Tridion has a slick, highly elaborate interface, which was updated in 2009, and then completely overhauled again in version 2011.

On the plus side, Tridion’s user interface is available in multiple languages (including right-to-left), with a multilingual spell check. Figure 148 shows an example of the navigation using Kanji.

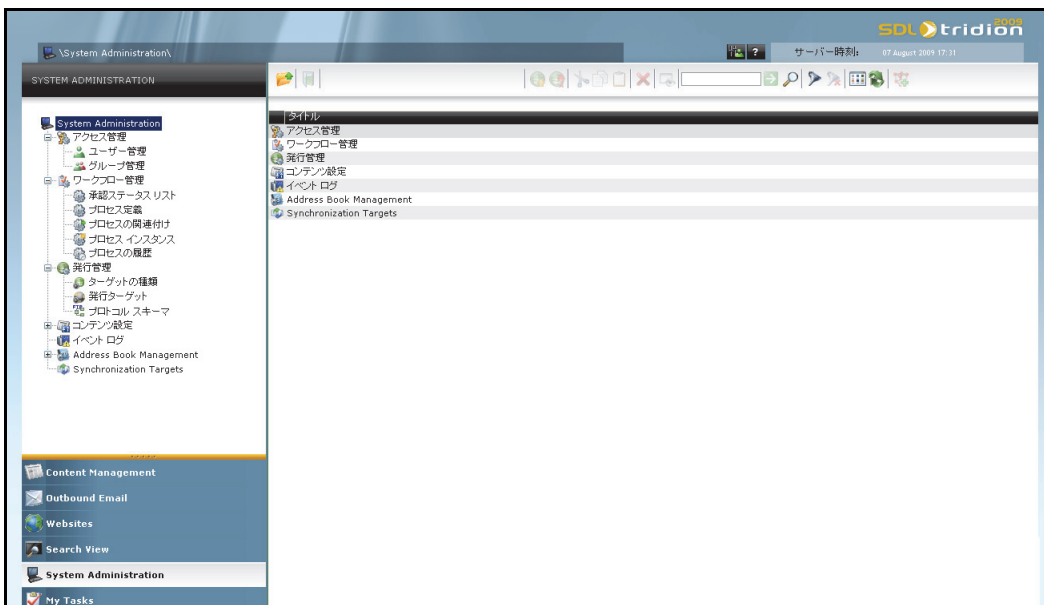


Figure 148. SDL Tridion’s interface supports non-Western glyphs, including right-to-left languages.

Contributing Content

Usability deficiencies notwithstanding, Tridion’s authoring services remain decent. SDL recognizes that companies and contributors alike reside on a functional maturity spectrum from basic to more complex. On the simple side, some users need only an in-context WYSIWYG publishing interface with highly structured templates. If you are familiar with SDL Tridion 2011 and earlier, you may remember their add-on module called SiteEdit that was designed to provide just that — a simple, in-context editing environment. Rife with bugs and core CMS compatibility issues due to consistently out-of-sync release schedules, SiteEdit since has been discontinued. You will find some of its functionality built in the new UI referred to as Experience Manager.

For more complex authoring tasks and more technically inclined content authors and marketing managers, SDL Tridion provides its traditional interface, where you can take actions like reconciling content from multiple repositories, selectively importing content from other enterprise websites, opening multiple content items concurrently in separate windows, and accessing various schemas.

SDL Tridion employs its own custom rich text editor called Format Area Control. It will clean up most of Word HTML and convert it to XHTML the way that other thick clients will do. You can also use WebDAV to check files in and out. This is especially handy for images; for other types of binaries, carefully test the outcome.

Administrators can define varying levels of accessibility checks on content entered into the system. These filters are defined using XML Schemas (XSDs) for each content type. For example, SDL offers filters for W3C compliance (according to three commonly accepted levels of strictness) using XSD from that organization.

Taxonomy management in the contributor UI makes SDL Tridion potentially more useful for product-oriented sites. There are some limitations to it (for example, not being able to use the same keyword in a multi-faceted taxonomy), which has engendered various hacks and workarounds. Content nodes can be added to several categories (i.e., a wine can be both in “red” and “Italian”), and the categories themselves can have metadata. Note, however, that since the functionality was previously hidden in undocumented API calls, experience with it remains relatively thin.

SDL Tridion stores XML components in arbitrary “folders,” and pages in “structure groups,” which are roughly analogous to site directories. Searching across both is improving with SDL’s switch from Autonomy (formerly Verity) K2 to the Lucene search engine. You’ll want to test this capability — particularly the results of the queries.

The company offers a generic media library that allows you to apply most of the product’s features (versioning, workflow, text search, deployment, etc.) to any binary files. Traditionally, SDL Tridion has not been very useful in terms of out-of-the-box image handling capabilities. Multiple consultingware add-ons have been crafted to address this weakness and to allow images to be resized, rotated, and sharpened.

Beyond that, SDL turned its acquisition of Media Asset Management vendor Calamares into an add-on module called Media Manager, which is available for a separate fee. Media Manager does more than manage images. The strength of this product is asset distribution to various media platforms and devices, including web, mobile, IPTV, and social media channels.

Keep in mind that former Calamares is not very well suited for light digital asset management tasks, as its core focus is primarily on video.

Content Lifecycle

For workflow modeling, your developers use a Visio plugin that will connect directly to the CMS server (Figure 149). After designing the flow, you can right-click and configure each step using a scripting interface. (Note, however, that “automatic activities” will have to be coded in VBScript and compiled into a DLL first, as with “events;” the system doesn’t come with pre-compiled DLLs for even the most basic functionality, such as sending a notification email or publishing an item.) This is handy, but arguably less usable for business users. On the plus side, the system allows for parallel approval flows of arbitrary complexity — a necessity in many scenarios that would involve translation.

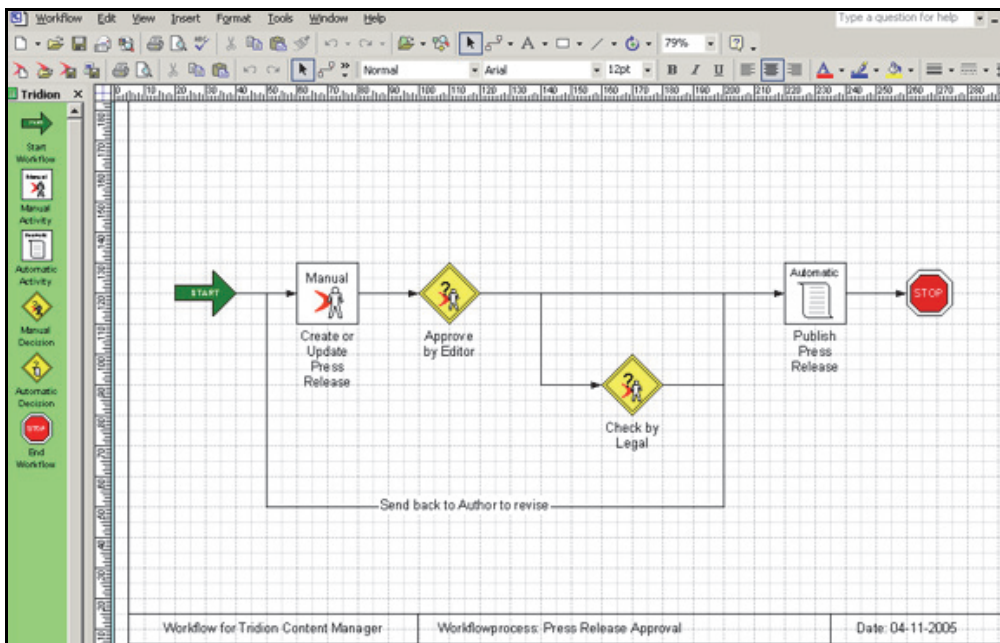


Figure 149. Visio plugin for workflow modeling.

From an end-user perspective, workflow functions are nicely integrated into the main control panel, which makes the approval process a bit more transparent. With the release of SDL Tridion 2013, there’s a new feature for workflow bundles. These bundles allow you to combine various elements that need to be approved all at once or in contextual correlation with each other into a package. Other useful workflow features recently introduced by SDL include escalation paths for overdue activities and workflow due dates that are visible in the UI. It is important for you to understand that these features are very new and have yet to be tested in many real installations. Test (and then retest) carefully.

SDL (the parent company) is heavily focused on translation management products, so it’s not surprising that SDL Tridion is strongly geared toward multi-language support and the kind of workflows you’d expect to see in an environment where translation and reuse of content across sites or microsites is key. This was actually one of SDL Tridion’s strong points well before the SDL acquisition. It no doubt weighed heavily in SDL’s decision to acquire Tridion.

The unique demands of multi-site/multi-language publishing have unavoidably driven the product architecture strongly in the direction of content reuse.

Using the “Translation Manager,” you can right-click to submit an item for translation from the CMS to one of the two available Translation Management Systems:

- SDL’s TMS
- WorldServer

Translators in TMS can work in their familiar environment and translate components. Upon submission, SDL sends the item back to Tridion for approval. It would be nice to see all this in one dashboard rather than two, but no other vendors have pulled this off either.

SDL Tridion also offers “Archive Manager” functionality, with the ability to have the system automatically capture deltas so that site content (rendered or raw) can be viewed at a point in time in the past. The deltas are captured incrementally, which means the system does not simply take a “snapshot” of the site every 24 hours (which can be expensive in storage as well as processing power). The “Archive Manager” has caught the attention of customers in pharma and financial industries in particular, where the ability to look at legally sensitive information exactly as it presented at any given point in time in the site’s history can have important governance implications.

Written in C# by SDL Tridion itself, Archive Manager exposes no APIs yet, so it’s something that needs to work out of the box for you, and because of this, we strongly suggest testing before you buy. The module is sold as an add-on and is priced on a per-site basis.

Experience

Publishing

Depending on how well you implement the system, Tridion can output relatively clean code with friendly URLs specified by authors. However, binary assets (such as images) must reside in a specific folder and will always contain an XML identifier as part of their name (e.g., “ChartPortfolio_CI_icon_tcm113-20455.png”). For expert developers, it’s possible to circumvent this issue.

SDL Tridion has paused its previous efforts to integrate its WCMS with print channels. If you hear a sales pitch that talks about its Adobe InDesign plugin, take that claim with a grain of salt, as this integration has not been updated for years. If print-to-web, or web-to-print is important to your organization, you may want to invest your own development effort into this plugin.

Experience	
Publishing	
Standards Adherence	
Multichannel	
Mobile	
Digital Marketing	
Site & Campaign Analytics	
Testing & Optimization	
Segmentation & Personalization	
Social Media Integration	
Promotional Campaigns	
Community & UCG	
Workplace	
Collaboration & Networking	
Dashboard	
Ancillary	
Site Search	
Online Forms	
Module Ecosystem	

Digital Marketing

SDL Tridion facilitates a variety of delivery features, most notably profile-based personalization in its Personalization & Profiling module, where you target content and email campaigns to individual segments, while excluding others. The content can then be published statically or, for dynamic and personalized content, can be retrieved using (for example) ASP files that make calls into an API that SDL Tridion provides. In the past few years, SDL Tridion has struggled to make sense of a set of its homegrown technologies in combination with those from acquisitions. Its long-standing Personalization & Profiling and Audience Manager modules have been combined (or, one could say, replaced) with an acquired product called “SmartTarget.”

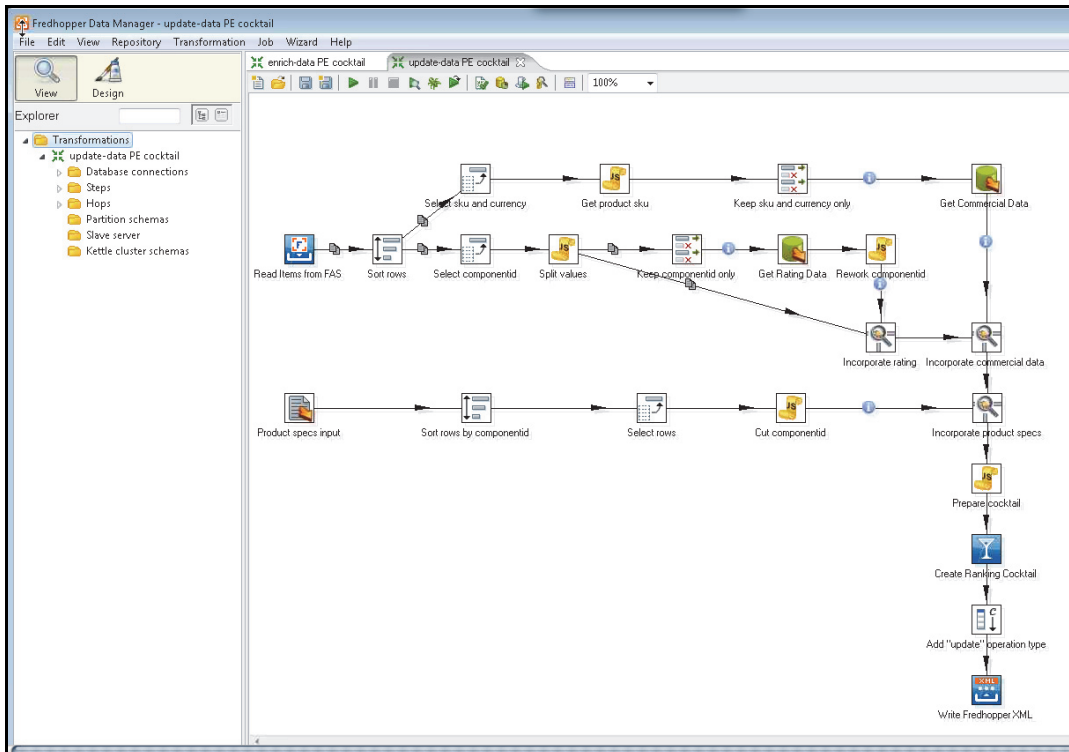


Figure 150. A view into what's under the hood of SDL SmartTarget based on the acquisition of Fredhopper technology. Fredhopper Data Manager is not for the faint of heart, and only highly technical editors or developers will be the ones dealing with the complexities of “data cocktails.”

In any case, SDL Tridion has been evolving experience and campaign management features steadily, which have helped make it a more all-encompassing solution for marketing managers, but it lacks some of the sophisticated simulations of Adobe, the e-marketing breadth of Sitecore, and its personalization services are not as sophisticated as Oracle’s.

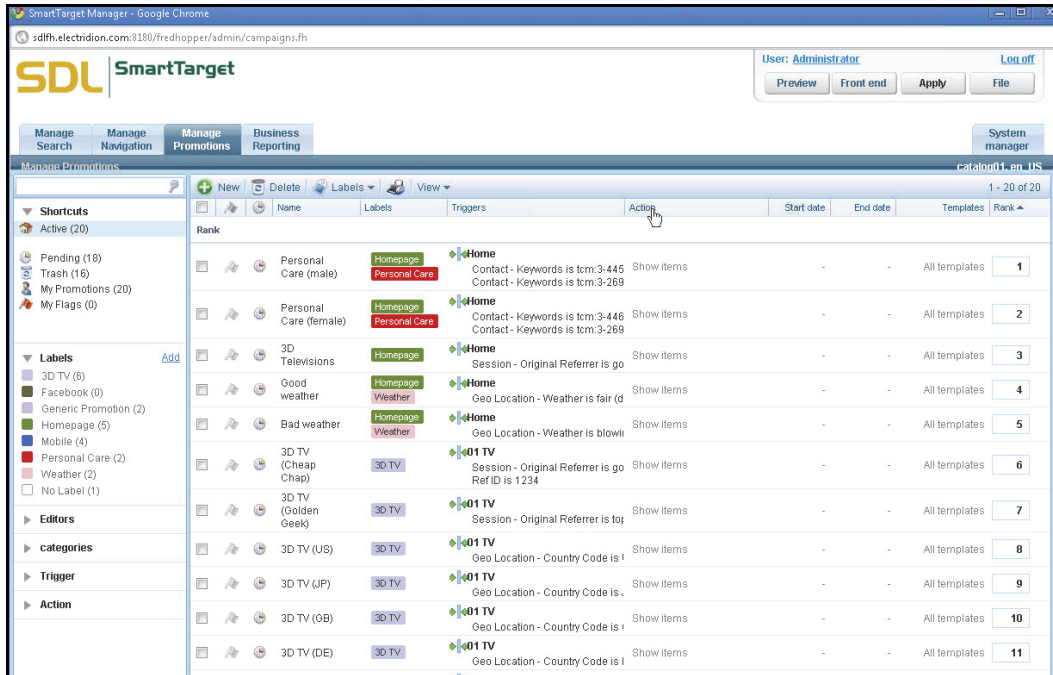


Figure 151. The SDL Tridion SmartTarget interface where marketing is supposed to work.

The Outbound E-mail and Audience Manager modules have become flexible tools for creating and managing email newsletter campaigns, with the ability to automate user management (from sign-up to opt-out), generate rich-HTML newsletters from web content, and do outbound mailings that can occur piecemeal fashion (over defined intervals, so your sales people aren't overwhelmed with huge waves of responses all at once), or shotgun-blast style. SDL Tridion has come up with some nice features, but once again, you should weigh them against your actual needs and test the features that matter most to you under conditions that replicate what you actually intend to do in production.

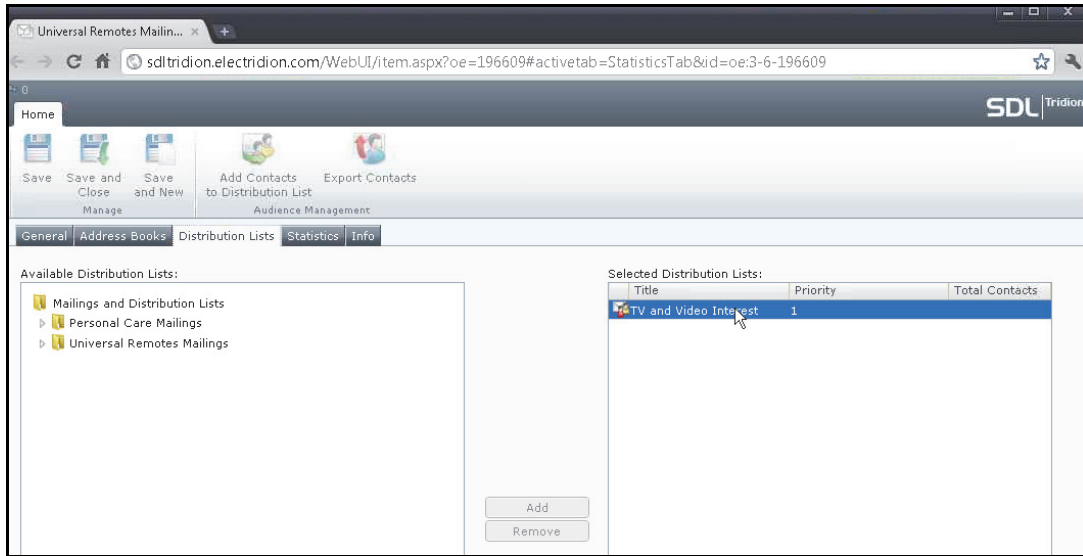


Figure 152. Mailing and distribution list management in SDL Tridion WCMS — targeting an email to a particular segment and distribution list.

Instead of what used to be known as the Communications Statistics, SDL Tridion is now shifting its focus to the so-called Online Marketing Explorer, which combines some campaign management and “initiative” tracking with analytics capabilities.

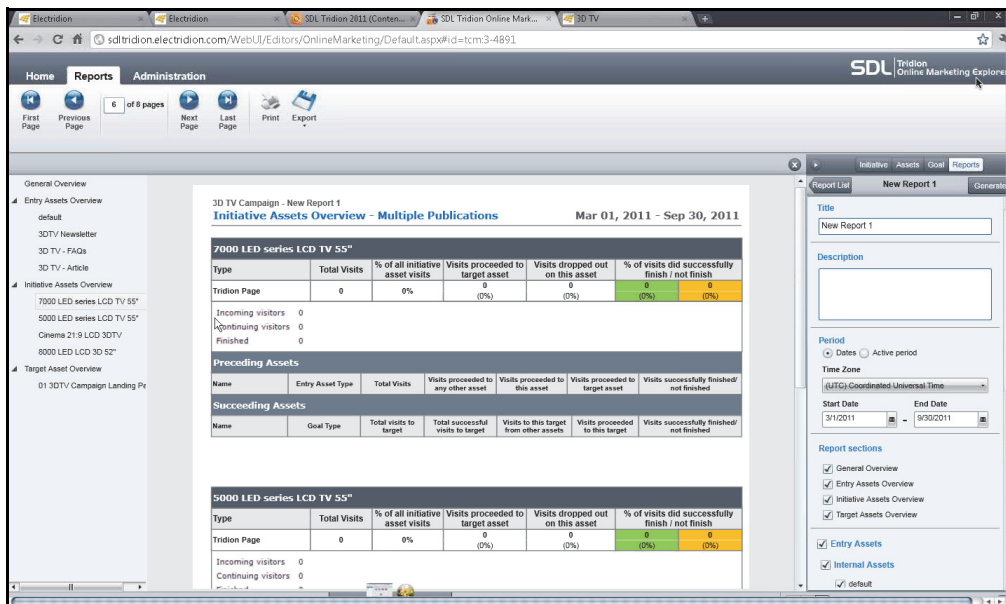


Figure 153. Yet another interface to master, the Online Marketing Explorer UI.

Note that on the analytics side, the company is still sticking to developing its own, native analytics, as opposed to opening the door to integrations with major analytics providers. While the latter is still possible, keep in mind that you might be getting two conflicting sets of data, at a time when the industry trend is to integrate third-party analytics tools. Also note that the Online Marketing Explorer module is priced separately from the overall package.

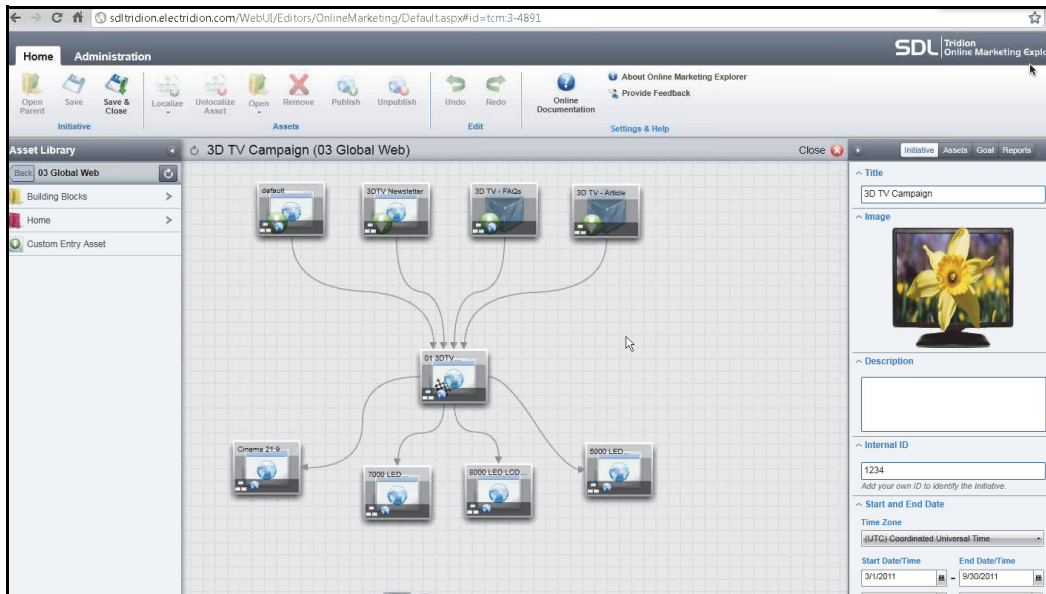


Figure 154. Campaign management in SDL Tridion Online Marketing Explorer.

Up until the 2011 release, SDL Tridion was surprisingly lacking in native “social” applications, like wikis. There are still no wikis, but some effort has been made to enable blogging platforms integration, as well as connectors to Twitter, Facebook, and YouTube.

When there’s social, there’s mobile. In case with SDL Tridion, it’s more of an attempt to show what’s possible rather than productizing specific mobile features. Here we are talking about an OEM deal with Netbiscuits. According to SDL demos, you can (with some work) have content published from SDL Tridion get recognized and displayed properly by various mobile devices and tablets.

Ancillary Services

You should note that SDL Tridion’s delivery tier doesn’t support search out of the box. You will have to add a third-party solution. This can be problematic, since most search engines likely only will collect the content by crawling the resulting site; the highly componentized nature of Tridion sites has been known to trip simpler solutions like Google Appliances, particularly when content is reused or personalized.

Vendor Intangibles

SDL is a large company, with around 2,200 employees, of whom 200 work on the Tridion side (a number that has stayed constant in the past two to three years). While SDL’s growth has slowed down in the current economic climate, revenues year-over-year are still up. According to the company, about a quarter of SDL’s revenues are generated by content technologies. Boosted by these results, SDL continues to acquire software companies. After having added translation services companies, and then translation management software companies, the focus the past

Intangibles	
Vendor Services	
Vendor Professional Services	●
Channel Partner Services	●
Support & Community	●
Strategy & Roadmap	●
Viability & Stability	●

few years has been on XML and component management companies (including Tridion, Trisoft, and LiveContent from the XyEnterprise acquisition). The result is a product portfolio that is relatively comprehensive, with products from translation to component content management covering the full range of what SDL calls “Customer Experience Management.” At the same time, the individual products often compete for the same business, and overlap functionally. In essence, SDL is still largely a translation services company, with the revenue from those services fuelling expansion in traditional software markets. For now, when the company talks about “integrations” between their products, you should really think, “connecting.”

SDL’s focus has been shifting very slowly toward making the product attractive to modern-day marketers under the mantra of experience management. At the beginning of 2010, SDL bought Fredhopper (a Dutch e-commerce company, somewhat comparable to Endeca, which we cover in our Enterprise Search research). The Fredhopper acquisition has been worked into the SmartTarget product, but the integration remains rather shallow. It will take SDL considerable effort to marry the two technologies together in a meaningful way. Another recent acquisition of a Media Asset Management company is Calamares, which is being integrated as the SDL Tridion Media Manager in attempts to mitigate SDL Tridion’s dearth of native media asset management capabilities. Test carefully to determine how these two integrations operate beyond the nice demo.

The SDL Content Technologies division is becoming less of a European company, since it’s been working on expanding its sales and consulting presence in North America for the past several years. The company is also making efforts to conquer Asia, but so far, there’s little to report. SDL WCM’s partner channel has been growing outside Europe, with many European SIs creating subsidiaries in North America and following the largest amount of Tridion business across the Atlantic. However, experienced SDL Tridion developers continue to be in short supply — especially in North America, where the company has expanded its customer base a bit faster than its infrastructure can support.

At the same time, SDL remains the most authoritative source of expertise on the product, including the latest releases. As a result, SDL’s partners may be somewhat under-skilled in the newer products, at a time when SDL Tridion’s own consulting and support resources appear to be stretched. The company says it’s invested heavily in revamping training and support, and has introduced a partner certification program to improve field implementation quality. However, we’ve heard that training programs are long on PowerPoint slides and short on hands-on exploration of the product. Support is provided primarily from the company’s Netherlands headquarters, and is comparatively well regarded. SDL offers customer support from San Jose and New York.

In terms of product pricing, SDL Tridion’s licensing model is one of the most archaic on the WCM market. SDL still charges based on a dizzying array of variables, including the number of servers, CPUs, users, publications, and so on. This approach makes it hard for customers to understand exactly what they’re buying initially, as well as what the downstream financial implications will be when their implementations grow.

To complicate things even further, on top of the base WCM product license, you’ll encounter a myriad of separate add-ons and modules that come at an additional cost. The new Online Marketing Explorer, for example, starts at \$70,000.

Conclusion

SDL Tridion salespeople will sometimes confess that the company has focused less on web experience management and digital marketing than key competitors like Adobe, Oracle, and Sitecore have. Maybe that's OK. For most Tridion customers, it's not about marketing innovations that attract them to the SDL product, but rather the venerable BluePrint for multi-site management. On the e-marketing side, most enterprises simply do not have the internal expertise or capacity to leverage advanced features anyway.

Nevertheless, longtime SDL Tridion customers and partners would prefer to see the underlying platform upgraded to a more unified, modern set of core technologies. As with HP, you must evaluate carefully whether SDL Tridion's attractive feature set today merits the potential pain of some tough upgrades down the road. You also must evaluate SDL Tridion on its unusually dense technical complexity, and calculate the cost not only of rolling out a production system but also of maintaining it over time. This is especially critical with the spanking new 2013 edition, which the company has been rolling out rather unevenly.

Licensees with large internal development teams who have committed to learning the peculiar dimensions of the product tend to be SDL Tridion's happiest customers. Customers with scant internal development resources should realize that there is no quick route to success with the product, given its many moving parts and wide variety of technologies. As with Adobe, getting your system into production could take anywhere from six to twelve months; we know some that have taken longer.

In the end, you'll want to weigh this product's obvious multisite management capabilities against its relative complexity and opacity. You'll also find that SDL has a somewhat different corporate culture than some of its fast-and-loose competitors. As always, take the time to test competitively both the technology in the supplier, and in head-to-head proofs of concept.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Sitecore: Sitecore CMS

www.sitecore.net

Vendor at a Glance

Specsheet	Sitecore: Sitecore CMS 7.0 Summary
Geography	Global, but strongest in Europe and North America
What's New	<ul style="list-style-type: none"> • Changes in the licensing model combines DMS (Digital Marketing System) with core CMS • Multiple enhancements to the search functionality, such as boosting, tagging, and facets
Strengths	<ul style="list-style-type: none"> • Product is highly extensible and componentized, and can offer portal-like functionality akin to SharePoint • Strong localization bent useful for multinational environments • Solid Lucene integration enables advanced search functionality • Best suited for highly structured content and scenarios with substantial integration with other internal systems, especially Microsoft-based • Feature-rich, Windows-like interface will appeal to some end users • Company boasts a growing stable of third-party resellers and integrators around the world
Weaknesses	<ul style="list-style-type: none"> • Comparatively developer-centric: More of a platform than an out-of-the-box solution, with longer implementation times • Older versions do not behave well in Firefox; best performance is in non-IE browser Chrome • Workflow is more developer friendly than user friendly • Overly rich interfaces may confuse casual contributors • Multisite management services are weaker than direct competition • Comparatively poor taxonomy functionality may require workarounds • Product typically purchased through a third-party reseller or integrator, leaving the customer once removed from the vendor
Potential Fit	Global Enterprise, Multichannel Publishing, Advanced Marketing Portal
Unlikely Fit	Informational, Basic Digital Marketing
Compare To	Ektron, Microsoft SharePoint, SDL, EPiServer, Adobe
Operating Systems	Window Server
Repository	Databases: Oracle, MS SQL Server
Client	Browser: IE preferred; Firefox, Chrome, Safari possible
App Platform	.NET 4.5; Windows Azure, ASP.NET MVC
Licensing	Based on servers and users, with a variety of modules and packages. Median deal size is \$200K, and can go up from there
Ownership	Privately held

Summary

Sitecore began as a smallish Denmark-based vendor, but has expanded globally on the strength of its integrator-friendly, .NET-based CMS. Sitecore is broadly functional and ships with a variety of developer tools. Clearly, the company is doing something right, since it is expanding very quickly, but licensees should note that the overall product is something of a framework out of the box, and far from trivial to customize.

Consider Sitecore for advanced digital marketing, highly interactive sites, with fast-changing content and layouts. The product is less well suited to Simpler Informational Sites, and may be overkill for sites with mostly static content.

More so than perhaps any other vendor at this tier, Sitecore has ventured into the arena of experience management, with a broad variety of embedded marketing services that attempt to duplicate what you may already have through best-of-breed alternatives.

To be successful with Sitecore, you need strong internal .NET skills and/or trusted, experienced Sitecore partners. You may have no choice but to work with a Sitecore partner, since the vendor itself sells almost exclusively through consulting firm resellers.

Scenario Fits	
Simpler Site	
Informational	<input type="radio"/>
Microsites & Landing	<input checked="" type="radio"/>
Mid-Range	
Basic Digital Marketing	<input checked="" type="radio"/>
Mobile Site	<input checked="" type="radio"/>
Community Oriented	<input checked="" type="radio"/>
Complex Site	
Advanced Marketing Portal	<input checked="" type="radio"/>
Global Enterprise	<input checked="" type="radio"/>
Multichannel Publishing	<input checked="" type="radio"/>
Ultra-Large Single	<input checked="" type="radio"/>
Digital Workplace	
SMB/Departmental	<input checked="" type="radio"/>
Global/Enterprise	<input checked="" type="radio"/>

Introduction

When Sitecore CMS began in 1999, it did two things right. First, it focused on developing a reseller channel of Microsoft-oriented integrators, cared for, and fed them very carefully. A network of over a thousand Certified Sitecore Partners has sprung up worldwide (close to half of them in North America), resulting in a reputed 4,000+ client installations. The focus on a partner-channel model has allowed Sitecore to keep its pricing consistent from year to year, while enabling the company to continue on a path of rapid expansion.

The other thing Sitecore did right was to settle on .NET earlier than most of its competitors. Sitecore has been able to use C#, XAML, and ASP.NET to achieve tight Windows integration (at a runtime level, as well as a user-interface level), via a highly componentized architecture, one that now benefits from a cornucopia of third-party .NET component libraries and Sitecore-developed expansion modules.

In the latest versions, Sitecore has focused on becoming a better candidate for large-scale, high-traffic scenarios, resulting in some enhancements to usability, stability, and performance. The latest version of the core CMS 7.0 came out in August 2013, with a focus on optimizing and fine-tuning the search engine.

Sitecore offers other products in addition to the core CMS. In June 2011, Sitecore released its Digital Marketing System (DMS), with a plethora of goodies for personalization, analytics, and engagement automation. Customer adoption of DMS has tended to lag, however. Sitecore initially sold DMS and CMS separately, but in 2013 combined both into an offering called

Customer Engagement Platform (CEP) and now licenses only bundled CEP — whether you need digital marketing technologies or not. Still, very few licensees seem brave enough to express more than curiosity about CEP.

CEP includes various homegrown modules, as well as those based on acquisitions like the 2011 Pectora acquisition for web-to-print services.

Sitecore’s 600 employees are spread amongst its Danish headquarters and sales offices in San Francisco, Boston, Toronto, Brisbane, Sydney, Copenhagen, Stockholm, London, Bremen, Munich, Tokyo, Hong Kong, and Amsterdam. Over a hundred of these staffers are based in North America.

Technology

Technical Administration and Security

Sitecore is a bundled production and delivery system, with a frying model for content assembly. You will need a full license for the CMS, and separate runtime licenses for all delivery servers. A few existing customers still run everything on one server, and you still can; however, most new implementations employ multiple servers. It is quite common to see customers run a CM plus two load-balanced CDs and a SQL Server Cluster. What that means is that you need to account for infrastructure costs for this setup, as well as additional resources to scale upward.

Sitecore stores content in a relational database, typically Microsoft’s SQL Server (although Oracle is officially supported, as well).

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

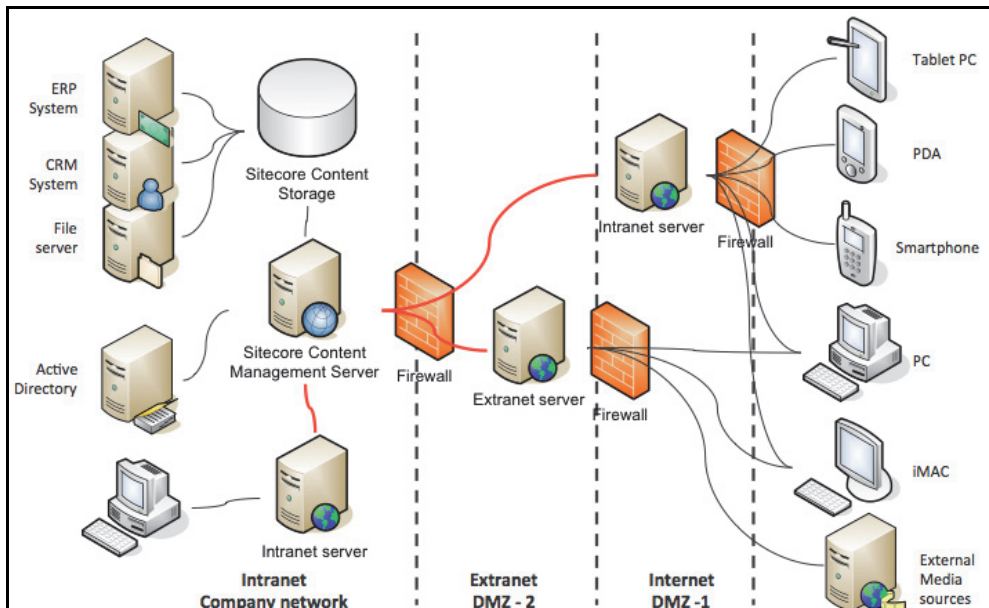


Figure 155. Sitecore's extended delivery architecture. (Source: Sitecore.)

Architecturally, Sitecore features a modularized approach to Content Delivery (CD) and Content Management (CM). Thus, these instances can be “pulled apart” more easily to run on separate infrastructures, but you still need Sitecore on both ends. The database takes care of persistence; the system now has an “event queue.” CM servers signal changes (e.g., when content is saved) via this queue, which the database then propagates across the other servers. Media files are replicated using Windows’ Distributed File System. Media files can be stored in binary “BLOB” formats in the CMS database and can be published to “CD” servers like other binaries.

It is possible to have multiple CM environments for editors in geographically disparate locations; e.g., one in the U.S., one in Europe, and one in Australia. Since the link between an editor’s browser and the CM server is bandwidth intensive (see below), this can be a boon to large-scale deployments.

Likewise, you could deploy delivery servers across the globe upon demand. These changes were necessary to enable Sitecore to work with Microsoft’s Azure cloud hosting. Instead of standing up your own Content Delivery servers, you can now publish to Azure. Sitecore has built a nice graphical interface for this, which makes it easy to set up new Azure deployments. However, few customers seem to take advantage of this route, perhaps because Azure is comparatively quite expensive.

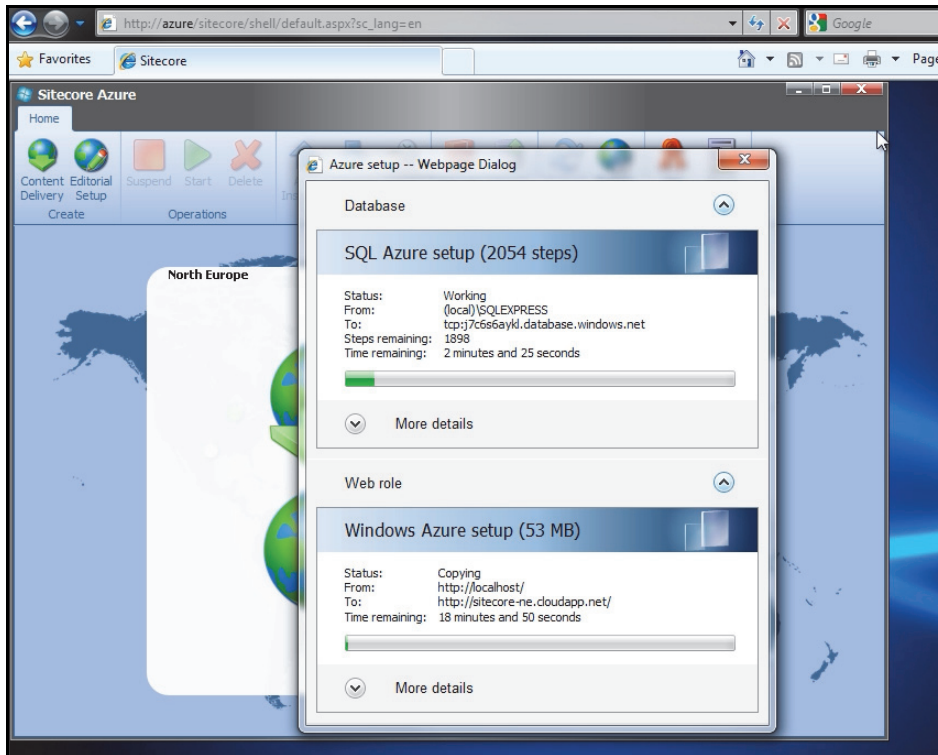


Figure 156. Setting up a new Content Delivery server on Microsoft's Azure cloud is, in theory, a simple task using Sitecore's interface.

In theory, this new plumbing makes Sitecore a more viable alternative for large-scale deployments. In practice, there are several ways the event queue mechanism can break. A lot will depend on your database, which is now burdened with the all-important task of keeping different servers in line. And, although your CM servers can be located anywhere in the world you want, the database will still rely on more traditional clustering. You may very well move the bottleneck from client/CM communication to CM/database traffic.

Sitecore distinguishes between its WIP database (containing all versions of all content, including unapproved/unpublished data) and the live-site database (containing only approved and published versions of each data item). The architecture also includes a “data provider” API layer to allow integration of the primary WIP data store with external providers, such as SharePoint, other company databases, or almost any remote system for which you want to write custom code. When content is published, data from any external provider is replicated to the live database — making it native to the content delivery environment. A data provider may be read-only or bidirectional.

Content deployment is straightforward, and can be done manually or on a scheduled, automated basis with the Publish Agent. You can deploy content on a per-item basis, in a packaged .zip file, or push the whole site. It's also possible to transfer content individually from the work-in-progress database to the delivery-server database, using wizards designed for the purpose.

When packaging multiple heterogeneous items, there's at least one downside to consider. Depending on the type of changes, you may have to endure some downtime due to the reboot

needed for Lucene to re-index fully. This is especially applicable if you're making changes to the content model. With enhancements to the search engine made in version 7, theoretically you could avoid this unpleasant occurrence via a manual override in favor of an incremental index that affected only the branch with structural changes. Alternatively, you can invest in more hardware so indexes can be rebuilt by a dedicated indexing server before deployment, and then transferred to a delivery server to reduce downtime. Since Sitecore is fundamentally a frying system, it has a built-in logical staging environment.

If you look at Sitecore from a traditional DTAP perspective, you'll find that the CMS is not very sophisticated when it comes to deploying content and/or code across environments. When pushing from development to testing environments, for instance, you can use Sitecore's built-in packaging tool to deploy your new code up the chain. However, when it comes to syncing up your production content with your Dev environment you're left with clunky and painful database backups and restores; there's no easy way for content to flow back down the DTAP path.

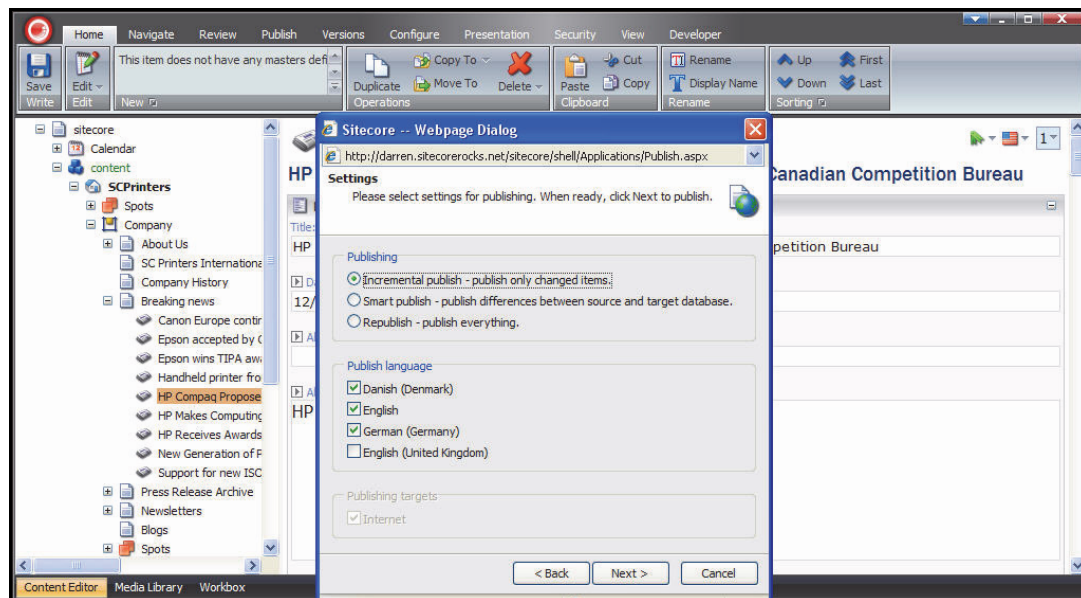


Figure 157. Publish settings can be configured on a per-content type basis. A Publish Agent facility allows items to be republished automatically at fixed refresh intervals, if desired.

However, there are quirks that some customers find unsettling. For example, Sitecore treats CSS files as code, thus requiring them to be packaged and deployed separately to delivery servers. This is unorthodox compared to a more traditional approach of treating CSS files as “assets” that are simply published to delivery servers (often by designers) without the need to resort to developers. With Sitecore, this approach may hinder agile development methodologies you've instituted.

Sitecore has sophisticated facilities for managing users, groups, and permissions based on the Microsoft's Service Provider Interface (instead of Sitecore's proprietary directory), and you can use multiple domains. Once users and/or groups have been registered with Sitecore, access control rights can be administered at an extremely granular level, using a check box-based user interface (Figure 158). It's possible to define custom “roles,” representing particular constellations of rights (read, write, create, publish, delete, administer) on particular asset

types, and map groups to roles or vice versa. If need be, rights can be customized on a user-by-user basis (and significantly, the same user can have different permissions across different sites). Interestingly, roles can also be nested (roles can be a member of other roles, and inherit those permissions.)

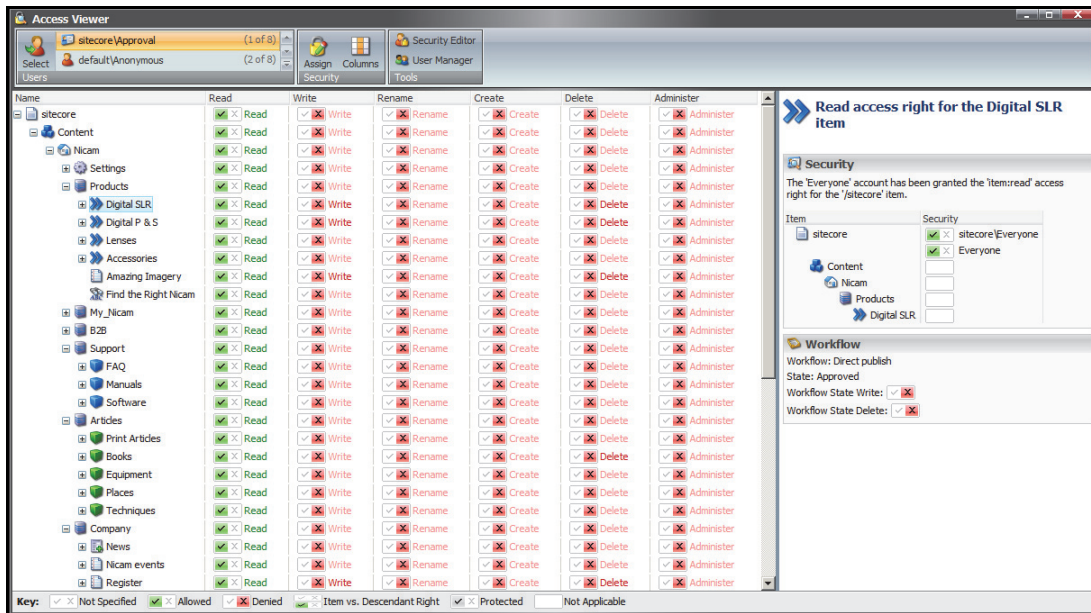


Figure 158. On a single screen, access rights can be set on particular assets across an entire site. In this example, it's for everybody belonging to the “Minimal” role.

As you can see, rights management is sufficiently complicated so that a novice site manager could easily get into trouble. Fortunately, the product will display trees, clarifying the effects of the layered access rights, so an admin can troubleshoot unexpected results.

One shortcoming of the product is in system reporting. The logs are all there, but making sense of copious quantities of log data is a challenge. You can remedy this by using the “Advanced System Reporter,” which can report on content items (i.e., items that are locked, or are about to expire), and users (recently created content, by author). You should be aware however, that this is a “shared source” module — that is to say, it’s free, but buying support for it is an optional extra. Alternatively, you can use the reporting functionality of the Digital Marketing System product, which was designed for reporting on visitor statistics, and can be used to analyze other Sitecore data.

Development

In terms of base technologies (ASP.NET, XPath, XSLT, and XAML), this platform is quite sophisticated (read: complicated) and diverse. Typical buyers rely heavily on Sitecore resellers or consulting partners for customization. At the same time, experienced developers typically express affection for the product, citing its flexibility and developer friendliness.

Like its competitors, Sitecore has its own lexicon for concepts surrounding templates and documents, although the terminology resembles Microsoft’s. The company calls structural content types “templates,” but these are not design templates and content types in Sitecore can be bound to multiple different displays. Sitecore calls an output template a “layout.” It is

comprised of “placeholders,” which in turn can contain various dynamic components and “renderings.” A rendering is produced when an XML record is transformed by the appropriate XSLT or .NET web control for display. As you can imagine, development for Sitecore is not a matter of asking your designer to produce the HTML templates.

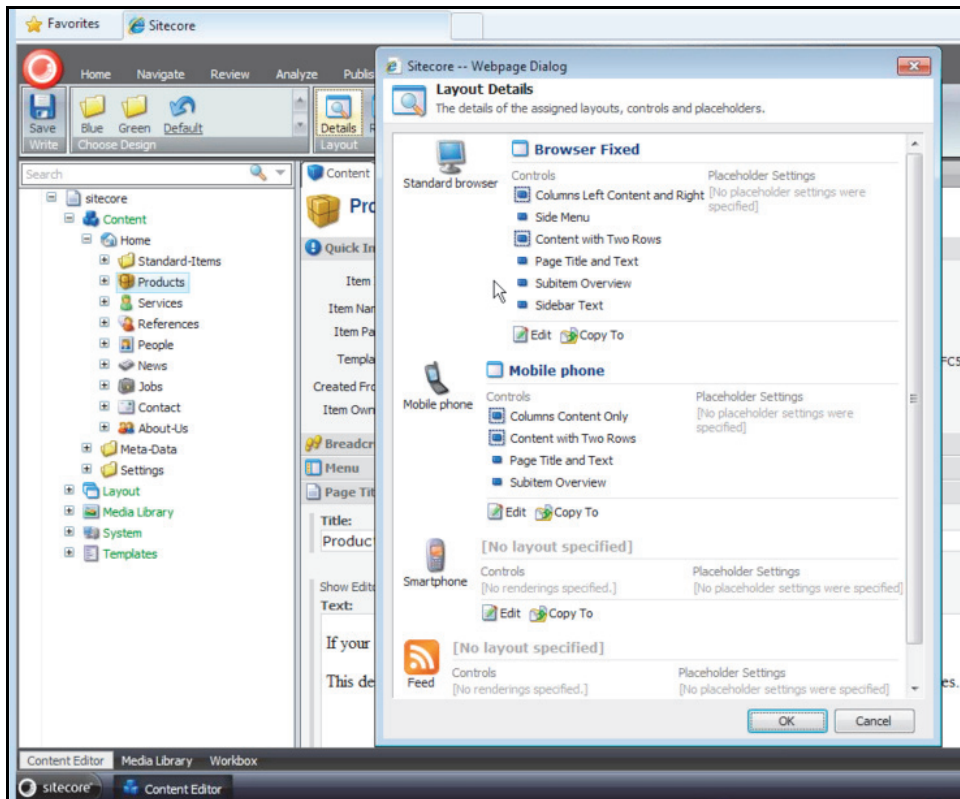


Figure 159. Managing layout details for browsers and mobile devices in Sitecore, after those specific layouts have been created by developers.

Sitecore’s integration with Microsoft Visual Studio is useful for developing ASPX template components, ASCX subcomponents, and XSLT style sheets. In general, Sitecore will appeal to developers well versed in Microsoft’s languages, and seasoned Sitecore developers will likely tell you there is much they can achieve in the system. You should be aware, however, that the learning curve is steep. Even seasoned developers cite poor documentation, shallow training, and a hard-to-navigate Sitecore Developer Network — all of the elements that normally would be the go-to resources for those learning a new CMS.

For content definition, Sitecore comes with a built-in layout editor for simple template editing and content type development (Figure 160). It’s not for the casual manager, but a trained administrator can make modifications without having to drop down to the code level.

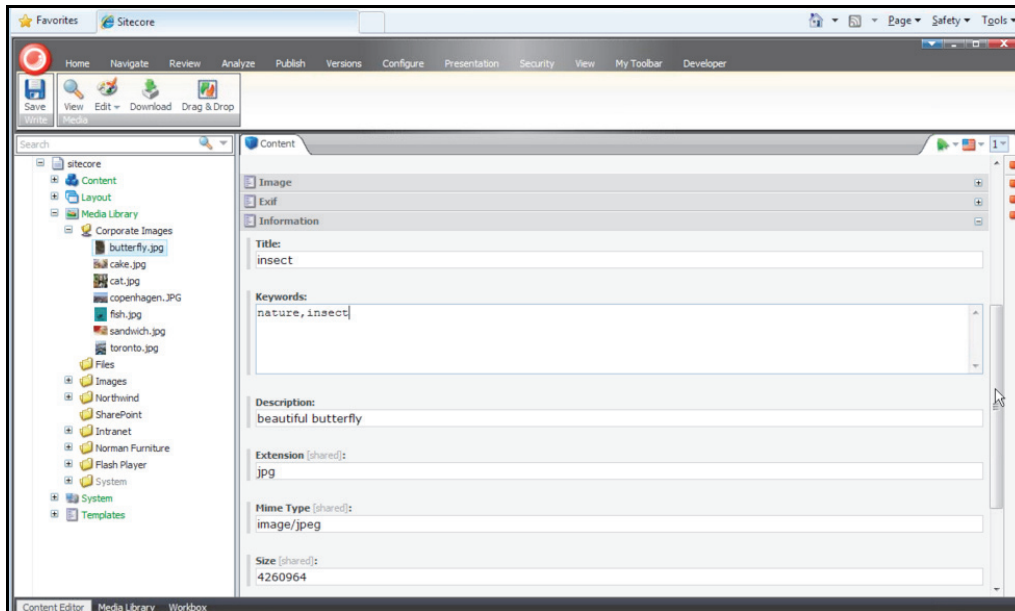


Figure 160. Image as one of the content types in Sitecore CMS. Experienced users can create new content types in the UI.

Unlike most of its direct competitors, Sitecore has a reasonably robust aggregation model. A rendition can consist of a dynamic call to a remote repository using an XML data provider function built into the product. Several integrators have developed connectors for SAP, Lotus Notes, and other repositories. Sitecore is highly componentized, achieving much of its functionality through pluggable modules and .NET components. For example, RSS support (which is bidirectional: Both consuming and producing) is provided through a separate module, as is LDAP integration, SharePoint integration, Word interoperability, Mailing List functionality, Calendar functionality, and many other functions. This makes Sitecore somewhat better suited to integration scenarios than most mid-market, .NET-based products.

As a general rule, you should view Sitecore as a labor-intensive development platform. While the product has commendable adherence to .NET (e.g., master pages, ASCX controls, web projects in Visual Studio, etc.), you will find that not all of its development is clean and straightforward. For example, various controls tend to commingle HTML and logic — a very messy way to code. In other instances, you may perform more raw coding than you would in competing products. Add that to less systemic abstraction and the often hard-coded elements, and you will understand why Sitecore-experienced developers are globally in such high demand.

Performance

As a frying system using XSL transformations or .NET web controls, Sitecore has had to pay careful attention to caching. The company decided to develop its own caching regimen in .NET. If good performance in the face of high HTTP request volume is a concern, allocate serious resources to testing.

Administrators can set caching rules at the placeholder level, and indeed, individual renditions (but not pages) get cached and then flushed when modified. We would urge you (as always) to

test an actual system against your projected traffic spikes. Happily, Sitecore provides some profiling aids, which include the ability to review rendering performance down to the component level (Figure 161).

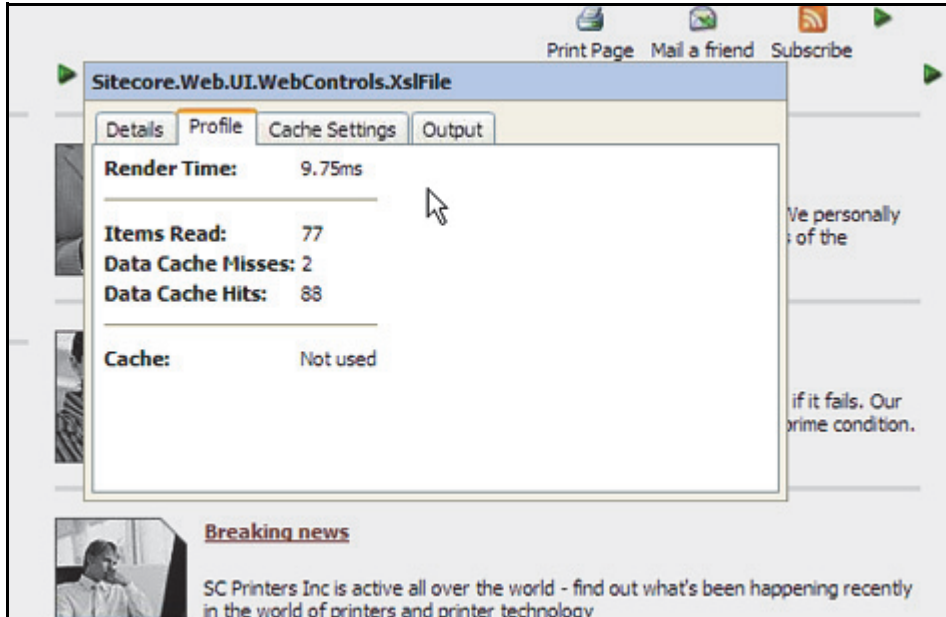


Figure 161. Performance profiling is available on a per-element basis, showing not only how long a given item took to render, but also how many cache hits occurred.

You'll want to do some basic performance testing against your contributors' desktop configurations, with relevant modules installed (such as the SharePoint module, if that's what you'll be using), to see if resource consumption could be a potential problem. The product has a reputation for behaving sluggishly. To be fair, this situation has been improving as organizations upgrade their hardware and move away from older, under-powered client machines. The important thing is to test the product in your own environment, with a variety of representative machines, using the types of integration modules you're likely to use.

There are options to improve both the responsiveness of the CM servers for editors, and the CD servers for visitors, such as using the Azure cloud on demand. Another option is to check-out managed hosting options via Rackspace, which customers report is reliable and responsive.

On the search engine side, there are plenty of opportunities to optimize performance, as well. As of version 7, Sitecore implemented a few new features to address shortcomings with search performance for its Lucene .NET implementation. Sitecore has caching of parent-child structures and makes all content indexable. Furthermore, you can select specific site sections to be search cached. Optimization occurs during crawling due to the implementation of a multi-threaded crawler and lookup tables, which improves crawling speeds. However, the horsepower of your own servers may limit some of these enhancements. Some customers put Lucene on a separate server; according to Sitecore, the search engine will support horizontal scaling, so you can add more search servers as needed.

Content

Contributor Experience

The product’s polished, JQuery-powered (and at times, AJAX-powered) browser interface looks and feels very much like Windows and MS Office. Such close adherence to Microsoft look-and-feel principles can be a blessing or a curse. Some usability experts favor mimicking Microsoft interfaces and the (in)famous ribbon on the grounds that everyone is familiar with Windows metaphors. Others have found, however, that Windows-like interfaces can confuse authors, or may not suite those favoring more elegant Mac-like interfaces. The bottom line as one customer said, “The UI is convenient, but only for computer-savvy users.”

Since the UI hasn’t been refreshed in a few years, it may feel dated. The clash becomes particularly clear when you see modern features implemented in an archaic user interface with the aging ribbon and iconography, ridden with popup windows that must be close manually. However, there’s one UI change implemented in version 7: the introduction of facets and tagging in the user interface as illustrated in Figure 162.

Content	
Contributor Experience	
Overall Usability	●
UI Accessibility	○
Contributing Content	
Authoring & Transformation	●
Tagging & Taxonomy	●
Content Reuse	●
Media & Document Management	●
Repository Services	●
Content Lifecycle	
Workflow	●
Globalization	●
Archiving & Compliance	●

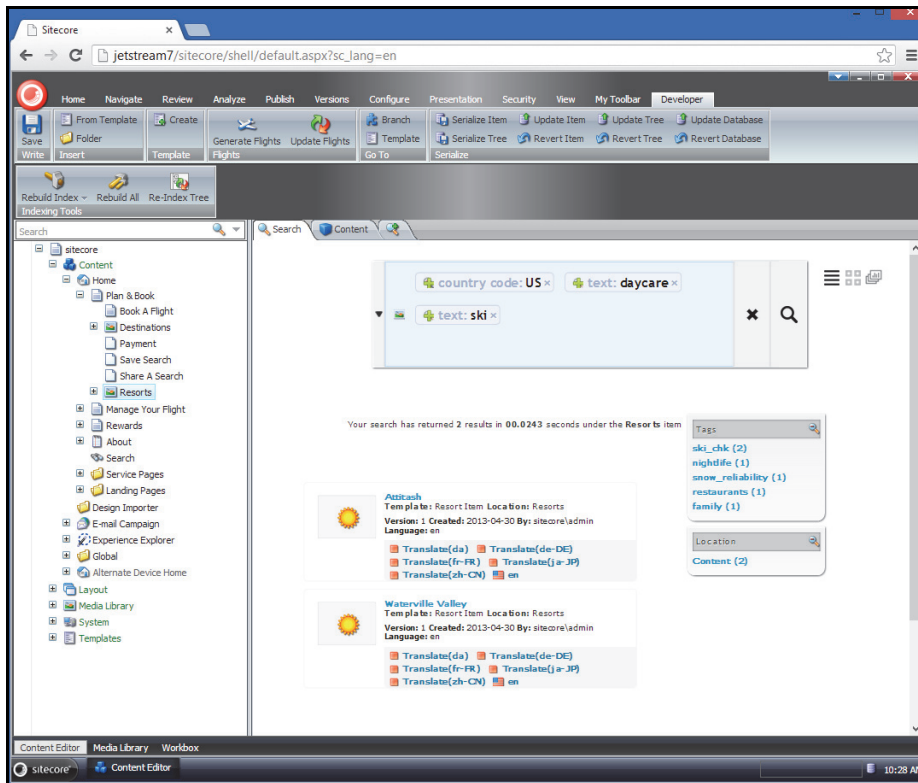


Figure 162. Version 7’s UI implemented tags and facets, which are designed to make Sitecore’s contributor experience less complicated.

Sitecore’s hope here is to liberate content contributors from a tree-driven approach to a search-driven way to manage content. If you had to pay close attention to parent-child relationships before — and many Sitecore users were overwhelmed by the sheer complexity of that and failed to do so — you now can browse content by tags and topics. This small enhancement may or may not be appreciated by content contributors. More advanced editors may welcome the change, while others may find it simply a quick patch to remedy the shortcomings of an overly complex interface.

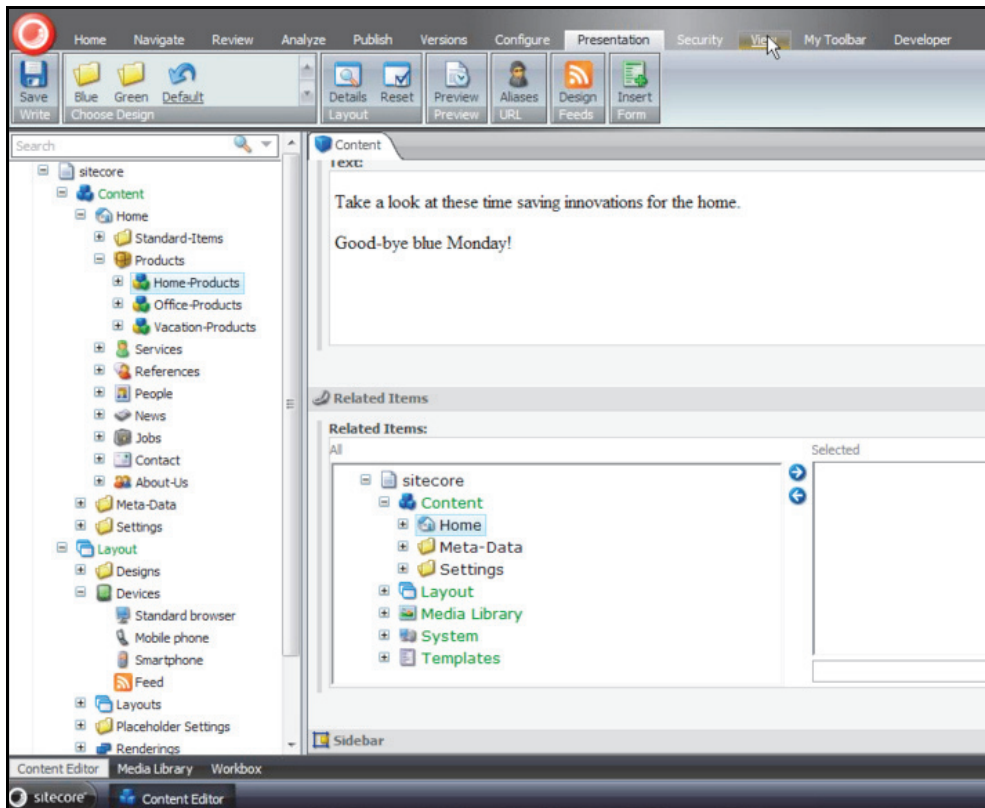


Figure 163. Assigning related items to a content item in Sitecore. This is where Sitecore developers often struggle with link resolver issues, but the editorial process looks fairly straightforward.

For WYSIWYG editing, Sitecore uses Telerik’s AJAX-based RAD editor. Editors find this editor more feature-full and flexible, while developers are happier with a more standards-compliant markup.

Cosmetic issues aside, Sitecore’s web client can be intimidating in terms of the sheer quantity of dialogs, widgets, and controls it employs. The power-user control panel interface can be eye straining when you switch to content contribution mode.

Cosmetic issues aside, Sitecore’s web client can be intimidating in terms of the sheer quantity of dialogs, widgets, and controls it employs. The power-user control panel interface can be eye-straining when you switch to content contribution mode.

However, Sitecore ships with six default user levels, each with increasingly simplified interfaces. The most basic (“minimal”) role will only see a few buttons on top of the editorial screen, with the ribbon reduced to equally sparse options. Depending on the complexity of the

role, the amount of buttons and clutter in the ribbon will increase, as will the amount of control a user is allowed to have.

Make sure these initial roles match the functionality required by your team. Most likely, you will want to tweak them, and changing default settings will require developer work. You'll still want a cross-section of users to test-drive the Sitecore UI before making any commitments.

On the plus side, the Sitecore interface has been localized into more than a dozen European and Asian languages. You set the default language interface through the browser, and you can choose the language you'd like to see at log-on time, using a drop-down menu. Each user can be associated with a particular language, meaning that some editors can get everything in English by default, and others can work with a user interface in French.

Contributing Content

Authoring, content import, and interoperability with various authoring environments are among Sitecore's strengths. As mentioned before, there are optional integration modules for Microsoft SharePoint, and Word, among others.



Figure 164. The in-context editing experience is different for various levels of content contributors, depending on which rights and permissions are assigned. Some users will have more options in the ribbon toolbar.

As with competing products, Sitecore offers in-context editing capabilities for casual contributors. It's possible to set parameters of the elements on a page from the in-context editor (if properly coded by developers). Sitecore can also validate content, both via pop-up report invoked from the toolbar, and by colored markers next to items with problems. For now,

the criteria are mostly technical: whether content is XHTML compliant, if images are the right size, or external links resolve.

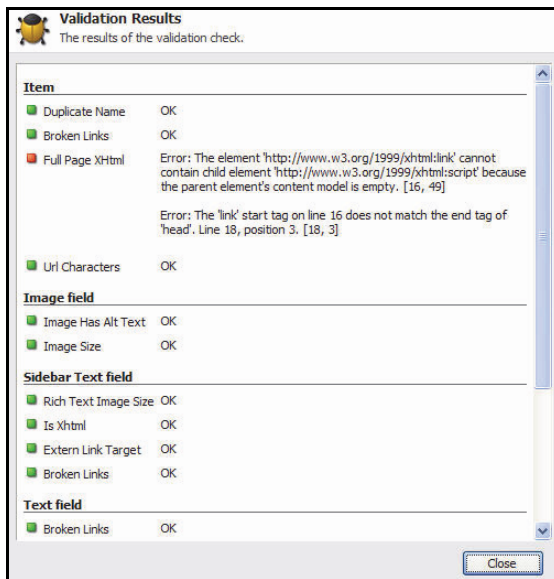


Figure 165. Validating against technical criteria.

Editors will find a comprehensive set of editorial features. For example, Sitecore will manage links as unique objects and alert authors when linked pages move. (Links are resolved at publishing time.) Editors can drag-and-drop content around the site (but the experience is less smooth than with Adobe Experience Manager, for example). The content in question can bring along its children with it — or not — as desired. The rich text editor can enforce image “alt” tags and table headers for accessibility. Editors can undertake side-by-side version compare. A server-side spell checker is available in 80 languages, and automatically follows the language in which the user is working.

The product’s Word import can map Word styles to particular CSS nodes. Note that by default, Sitecore performs minimal Word clean-up that is necessary for XHTML compliance. This means, in all likelihood, fewer user complaints, but messier code. On the plus side, Sitecore ships with a very nice W3C-code validation service (accessing W3C.org directly, via Web Services).

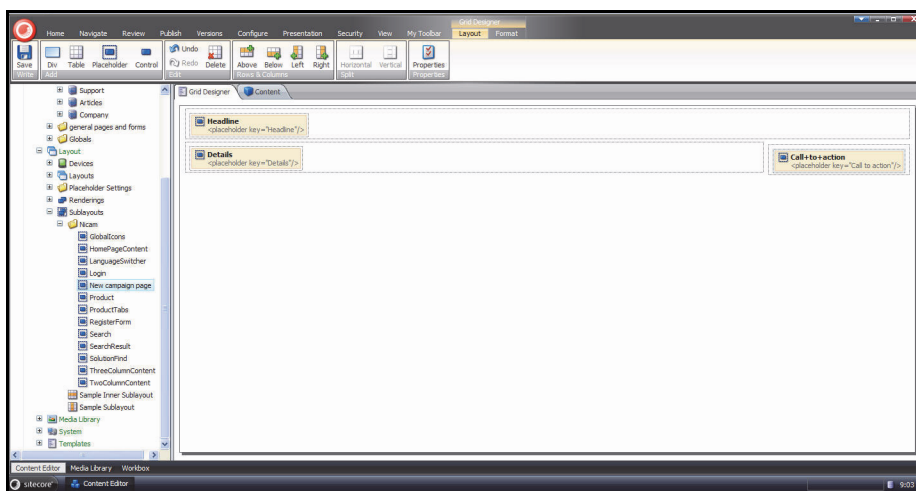


Figure 166. Building a new layout using the “Grid Designer.”

For page design and editing, Sitecore has a tool called the Unified Page Editor, where developers can define editable elements on pages. One of the ways to build presentation logic of pages is the “Grid Designer.” The Grid Designer is used to generate the page grids (for instance, “three-column page”) from the browser. These can be clicked together, with

additional HTML for more fine-grained control. Combined with the access to parameters from the in-context editor, this means that non-technical users could build new pages comparatively easily. However, as with comparable functionality in other products (such as mid-market EPiServer or Ektron), this relies on developers building the right controls and grids, first. You'll want to decide whether increased development time in the initial stages will pay off in flexibility for business users in the long term.

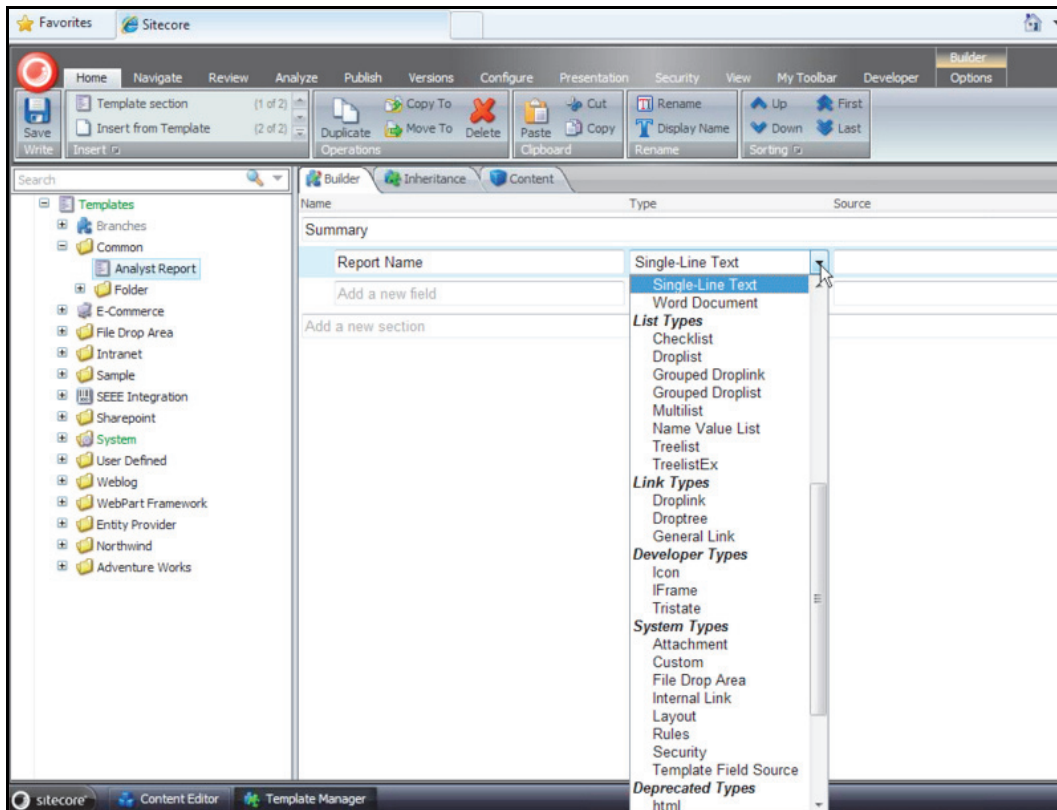


Figure 167. Sitecore's Template Builder: Some power users may use it, but this is mostly a developer-centric environment, where developers create elements for content contributors to use in the Page Editor.

Sitecore has the ability to create templates by “multiple inheritance,” meaning you can reuse and combine elements from multiple templates to create a new one. You can also pre-populate commonly used fields and lock them so they can't be modified, which helps solve some of the problems of less advanced users having too much functionality available to them.

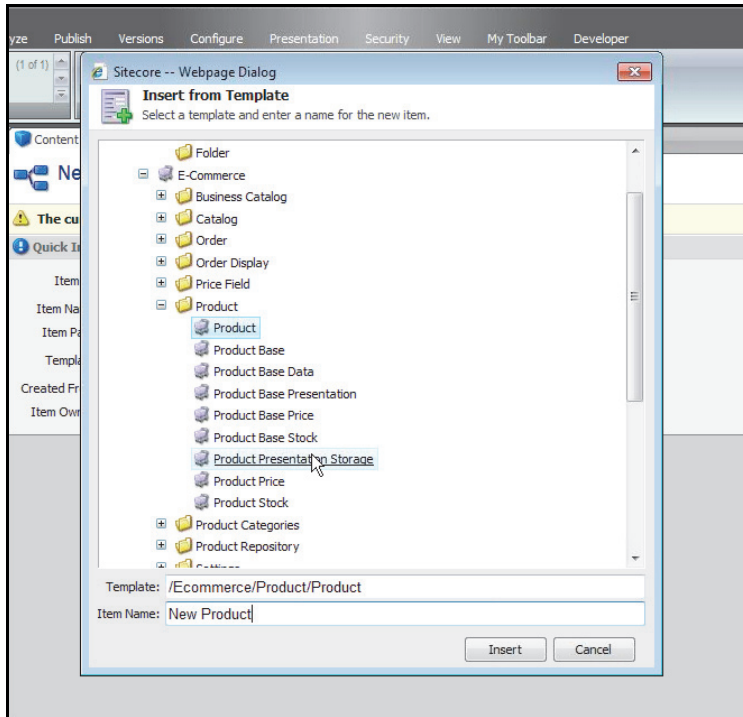


Figure 168. Sitecore can create new templates by reusing and combining existing elements.

Certain content types can even be hidden from certain groups of users. The “Branches” functionality will allow you to insert content (typically complete parts of a website or microsite) anywhere on a site (thus inheriting all of the original content and attributes).

The product’s repository services allows previews of any content item by past or future dates and times, as well as by roles and specific users. An added bonus is the ability to preview not just a page, but also an entire instance of the website at a future point in time. If you want to know how a specific user group is going to see the entire website in a week, you can preview and browse through the entire experience.

Sitecore has reasonably sophisticated asset management. It contains a media library, which supports versioning of video, images, PDFs, and other assets. The product’s integration with SharePoint allows access to image and document libraries stored on SharePoint. Once inside Sitecore, site managers can supplement existing metadata.

Content Lifecycle

Sitecore’s event-driven workflow system is competitive with those of other offerings in this tier (Figure 169). The basic building blocks are what Sitecore calls States, Actions, and Commands. You begin by specifying States, representing phases in the editorial process. Then you build Commands, and link the Commands to the States using Actions, which define transition points, and are equivalent to the lines connecting nodes on a flow diagram. Along the way, you can specify notifications, reminders, and archiving capabilities. In Sitecore, all editing operations start a pipeline (or flow instance) that advances content and other assets through the specific stages of a workflow. Each stage has an event hook that can be used to trigger actions outside the pipeline. Overall, the configuration system seems easy to use and more than adequate in building typical Web CMS workflows.

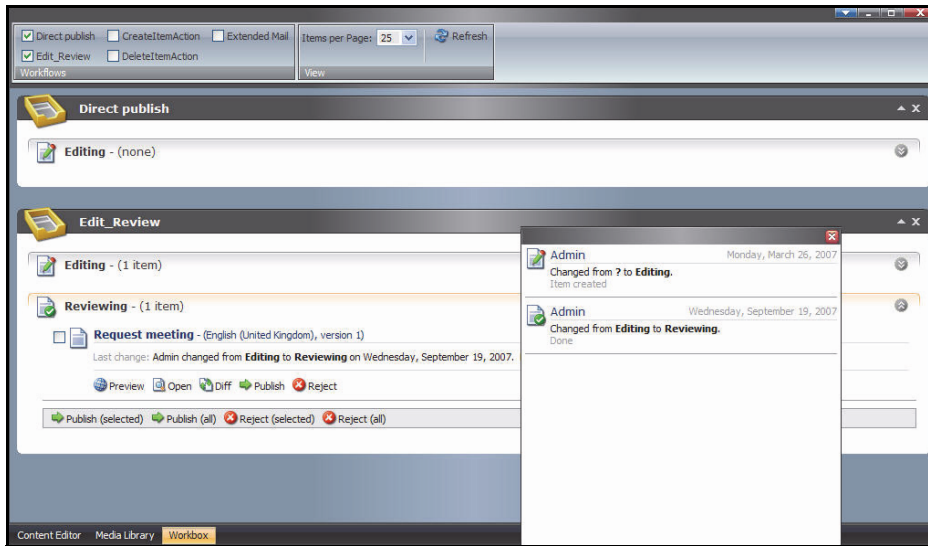


Figure 169. Workflow options are fairly sophisticated, but not always friendly.

Of course, the user experience of actually *participating* in workflows is another, more important matter. Sitecore does not natively allow you to put a collection of items through a workflow (though you can “publish” entire directories at one time). You can visualize version differences in the main control panel, but not within a workflow, where you might want to see what’s changed. Sitecore partners can customize all of this behavior at the API level, but that’s our point: You may need to invest in that to make the system more usable, particularly if you anticipate using a lot of workflow.

Although Sitecore excels at repurposing content, it is not particularly advanced with respect to true multisite management. It does allow you to “clone” sites (i.e., you can inherit both content and design into a new site), or create a “branch” of content items that you can use across different sites, but there is no dependency reporting, and no apparent signaling to local sites if a source item has changed. With time and painkillers, you can get a multisite implementation going, but it won’t be easy — or straightforward.

One final caution: if you expect your implementation to support multiple sites over time, you’ll want to bake that into the initial architecture and system configurations. Integrators report that it can be extremely difficult to convert an existing, one-site implementation into a multisite environment on the back-end. This is one of those places where Sitecore’s mid-market roots show through, relative to its more upscale platform competitors.

Nevertheless, the product performs capably at globalization — especially through its integration with ClayTablet. Sitecore can be set up to facilitate localization workflows, wherein authors can toggle between different language versions of the same content item, and submit content to multiple translator groups (though here again, the differencing visualization is not as strong as it is for other competing offerings).

With respect to retention, Sitecore comes with a modest built-in retention capability, such that site managers (or workflow processes) can rotate older content items into a separate “archive” database, where it remains under management without cluttering up the interface for current content managers.

Experience

Digital Marketing

Sitecore’s DMS (Digital Marketing Suite) is a module that is now delivered in tandem with the core CMS under the umbrella of a “Customer Engagement Platform” (CEP). You can no longer purchase them separately, after licensing changes that took place mid-2013, following the version 7 release. This approach may prove overkill for prospective licensees who may not be organizationally ready to delve into the digital marketing world and want to focus solely on the core CMS goals. Negotiate your maintenance fee baseline accordingly.

CEP is a complex offering that combines various old and new features under one roof, like engagement analytics, engagement automation, integrated email marketing, explicit and implicit profiling, and inline personalization (based on behavior, e.g., search terms), where you can specifically target these profiles.

Experience	
Publishing	
Standards Adherence	●
Multichannel	●
Mobile	●
Digital Marketing	
Site & Campaign Analytics	●
Testing & Optimization	●
Segmentation & Personalization	●
Social Media Integration	●
Promotional Campaigns	●
Community & UCG	●
Workplace	
Collaboration & Networking	●
Dashboard	●
Ancillary	
Site Search	●
Online Forms	●
Module Ecosystem	●

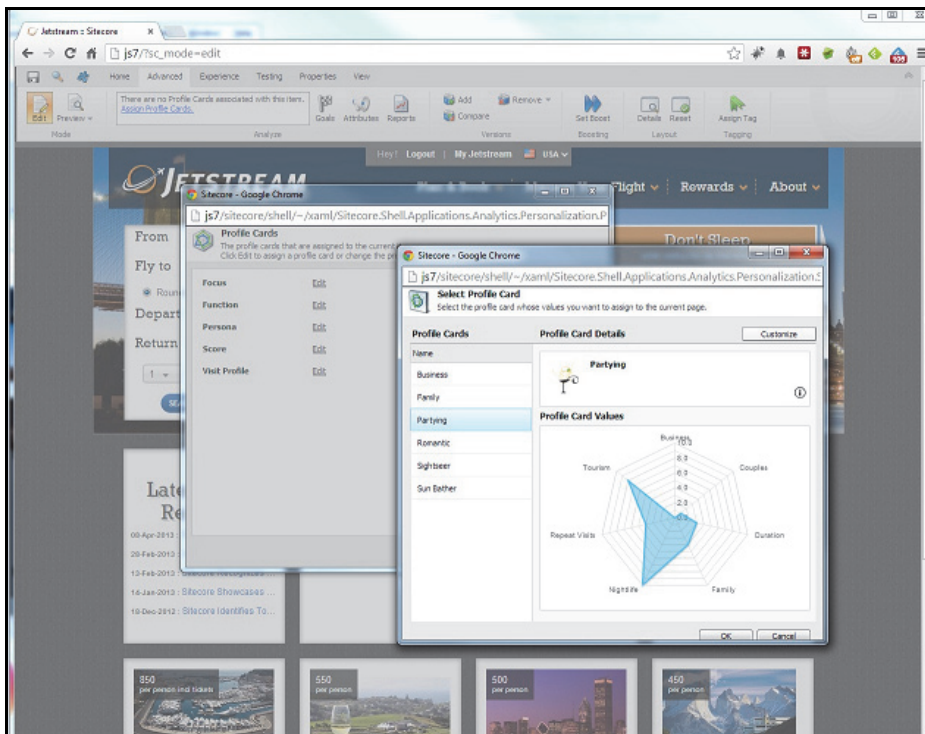


Figure 170. Inline profile management from within the context of a page in Sitecore.

With regard to engagement management (or many other functionalities that have been developed historically by Sitecore), the vendor tends to be ahead of real-life scenarios and actual needs of organizations. While the amount of depth and complexity in the engagement platform is worth an explorative trip if your company engages in heavy digital marketing activities, most organizations are not yet at the stage to utilize CEP or DMS.

The number of customers that have taken full advantage of CEP remain few and far and between. We've seen various attempts to implement DMS, but due to the relative freshness of the product and its overall complexity, implementations rarely go beyond a pilot phase. The vast majority of Sitecore customers and partners will need to spend a significant amount of time studying what they can potentially do with DMS before diving into any meaningful implementations.

Sitecore used to charge \$200K+ for DMS licenses, but then rolled everything into one CEP license that includes CMS and DMS. The price however, should be in the same range, depending on your requirements.

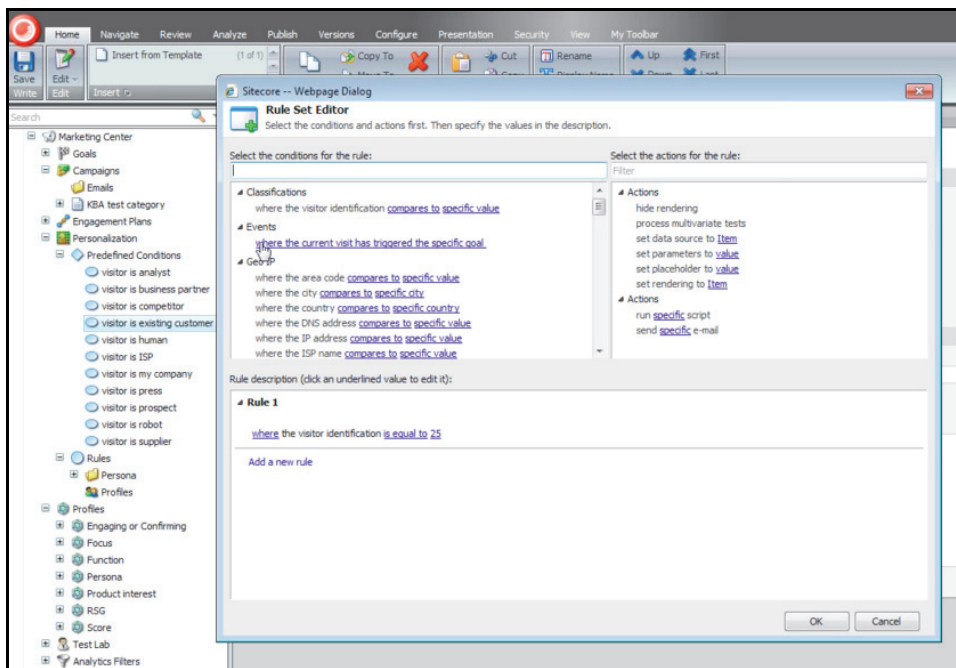


Figure 171. DMS rules management and creation as they relate to specific profiles set up in Sitecore — this can get very complex and granular all too quickly.

Here are some of the added functionalities you will see and should test-ride in the feature-expansive DMS:

- Digital Marketing automation through applying steps, conditions, and actions to engagement plans with monitoring and reporting
- Re-engineered API that opens more integration options — while making developers happy, this also adds a layer of complexity to an already complex platform
- Inline personalization, allowing marketers to set personalization values to content items within the editing interface, create rules, and preview them within pages

- Inline A/B and MVT for testing different versions of content for specific audiences
- Insights Dashboard, providing an accumulated view of the many moving parts in your running campaign
- Pattern engine that works like Amazon’s “customers who bought X, also bought Y,” based on a defined set of variables and predictive elements, feeding off user browsing behaviors and persona patterns
- Predictive analytics allow you to measure and target different segments based on behavior and patterns

By now, you may have noticed that Sitecore is designed for managing highly structured content. The system is therefore well suited for repurposing content for different channels, including mobile and print delivery. With mobile, you can build mobile-specific templates in Sitecore, and use the built-in WURFL-based library for device configuration and recognition.

With respect to analytics, the company has built its own metrics tool, which is included in the DMS. You shouldn’t think of this as a replacement for an analytics tool (if only because DMS isn’t well suited to analyzing large volumes of data); rather, this caters to editors and developers. The data can be used to create reports, which give editors feedback on how content is performing. Through a scoring system, DMS can determine how well certain visitor profiles are responding to the site. The data can plug into the personalization features, which means that (in theory) you can tailor the information to specific profiles as visitors click through the site.

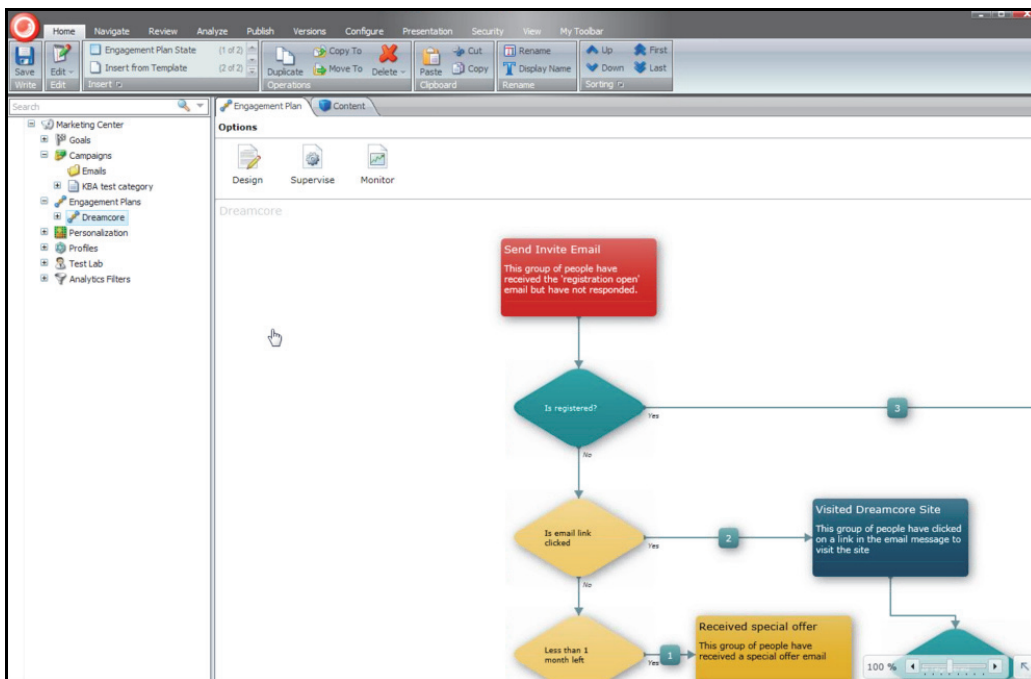


Figure 172. Graphical Digital Marketing Plan editor to create and edit lifecycles of customer engagement campaigns, including web and email touch points.

With analytics built-in to the platform, Sitecore argues that you should handle marketing automation within the CMS (as opposed to using best-of-breed tools, as most enterprises do). You can certainly explore this proposition, as there is some appeal here (e.g., there’s no need to tag every page with Eloqua tags). However, carefully investigate scalability in general, as well as Sitecore’s ability to support high email volumes, as well as more sophisticated services like drip marketing and deliverability support.

The analytics reports are advanced for a CMS vendor’s offering, but compared to full-fledged analytics, you may find the functionality limited. One big shortcoming is that the system only stores raw data, and not report data. An experienced analytics engineer will tell you that this model isn’t scalable on the storage or reporting side. Sitecore tells customers that it works just fine — on Sitecore’s own website. By now you should recognize a pattern with Sitecore: they have great vision on digital marketing, but a rather amateurish execution.

At the same time, you may find it useful for combined visitor session-level data sets with reports you set up in the CMS. You’ll almost surely want to employ an analytics tool to complement the built-in functionality. However (if you’re brave enough), you can try extending built-in functionality with the redesigned analytics reporting API that includes access to new engagement metrics.

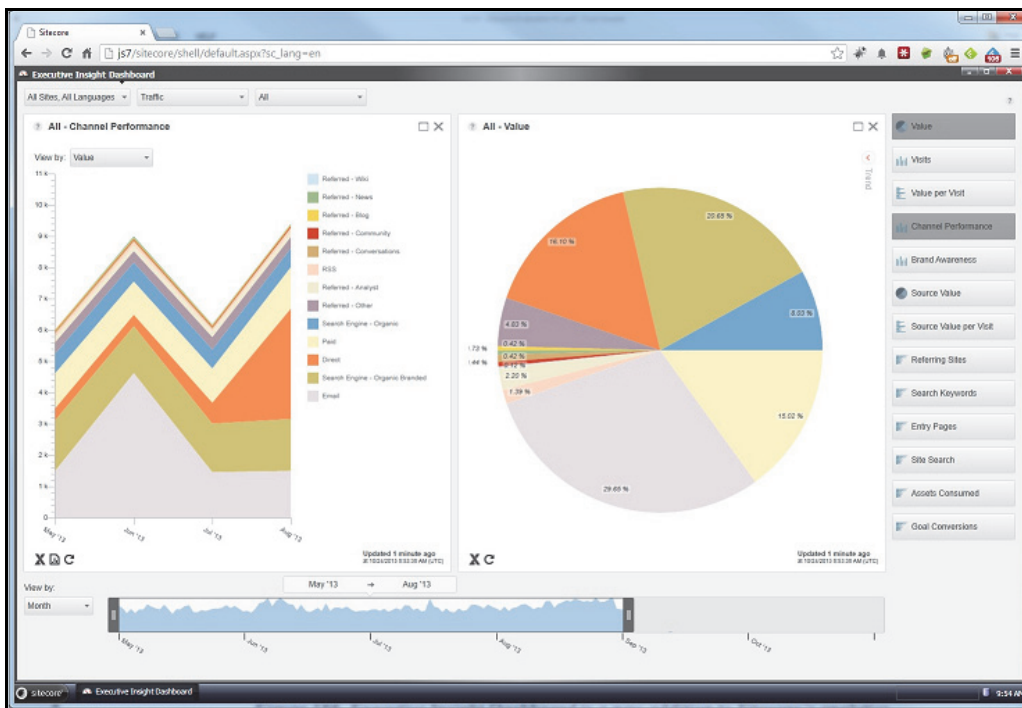


Figure 173. Executive Insight Dashboard is a new addition to Sitecore’s analytics package, offering a higher-level, bird’s-eye view on the complexities of site analytics.

In 2013 (in a friendly nod to its partner channel), Sitecore undertook a new approach to building applications on top of their core content management and digital marketing capabilities. One example is the campaign launcher application. It is a standalone app built on top of DMS and CMS, targeted to different marketing groups, and it is positioned as a perfect tool for Sitecore partners to get new business (or additional business) from their existing client

base. The campaign launcher is a framework for custom-built campaign management capabilities that can be customized for customers who are not brave enough to attack the DMS on their own on a large scale. Of course, this approach does require DMS licenses, because having just the CMS is not enough. You will need to access analytics, the campaign manager, and send targeted emails using DMS. To create campaign landing pages and upload any CSS, you will need the CMS.

A further step in this standalone movement undertaken by Sitecore is to make these apps cloud-friendly. In theory, this also allows partners to white label any Sitecore app and sell them as hosted services. Even if you have simple digital marketing tasks, carefully evaluate how these (fairly new and untested in the real world) apps work and what limitations you may encounter.

The Digital Marketing System is comparable to what SDL offers: stats for the editors, and hooks for personalization. However, Sitecore has a slight edge here. The analytics reporting is more advanced, and the personalization features are actually productized and easier accessible (SDL's have always remained in the territory of obscure API calls, with some improvements surfacing in the SDL Tridion release integrating the acquired Fredhopper). On the other hand, many enterprises have found two sets of analytics to become problematic, and some of Sitecore's competitors are successfully integrating third-party analytics tools in lieu of building their own.

Ancillary Services

For search, the product ships with Apache Lucene and Solr, the well-known open source search engine, for both internal and external search. Sitecore uses the Lucene.NET version, not the Java version. Alternatively, customers can purchase a third-party solution.

Sitecore offers a number of "micro-application" modules: Forums, Blogs, Calendar, Extranet, Newsletter, RSS, SharePoint connectors, LDAP, web forms, and more. Web forms, in particular, may prove to be too complicated for many editors. If the creation and management of web forms is mission critical to your organization, investigate it carefully in terms of capabilities, ease of use, and AJAX worthiness.

Sitecore has also attempted to improve its "out-of-the-box" capabilities by packaging features to meet specific enterprise needs. The "Intranet Portal," offering provides tools like blogs, forums, and calendars. For a centralized intranet — where the emphasis is on one or few departments, i.e. HR, controlling the content on the intranet — this may be a much more straightforward and cost-effective solution than SharePoint. However, beware of the fact that you then forgo SharePoint's extensive collaboration features. There's also the "Foundry" solution, which provides a set of tools for enterprises with many affiliates that manage multiple microsites. While Sitecore claims that these are "out of the box," this does not mean that you can dispense it without external help with installation and configuration.

Vendor Intangibles

According to the vendor, the company continues to see year-over-year revenue growth. As always, this is hard to verify with a privately held company, but the vendor seems to be doing well, with some 3,500 customers running a reported 30,000 websites on the software. Sitecore has been moving more aggressively into enterprise-level deals, and it is becoming increasingly common to see Sitecore competing head to head with SDL and Adobe in seven-figure procurements. Sitecore says the average deal size has increased in the past year, and the median deal is now \$200–250K.

Intangibles	
Vendor Services	
Vendor Professional Services	<input type="radio"/>
Channel Partner Services	<input checked="" type="radio"/>
Support & Community	<input checked="" type="radio"/>
Strategy & Roadmap	<input checked="" type="radio"/>
Viability & Stability	<input checked="" type="radio"/>

The company says its revenues break down to about 90 percent software, and 10 percent training and services. Indeed, some licensees are surprised to learn that projects are implemented by partners almost exclusively. In fact, most partners are resellers, and customers seldom work with Sitecore directly on implementation issues. What you may see in larger implementations is that Sitecore will parachute in one of its internal Subject Matter Experts (SME) to oversee the implementation or review implementation designs and code, but this is as close as you can get to Sitecore’s professional services, which mainly focuses on putting out fires and developing best practices to share with partners.

Partners typically provide first-line technical support during the implementation, followed by tech support from the vendor. Sitecore also offers various kinds of training courses that are developed in-house by the same SMEs.

Overall, your relationship to the “mother ship” cannot be as intimate as it may be with competing CMS vendors that rely less on partners than Sitecore does. Some customers express frustration when they try to get help from Sitecore and are continuously pushed off to third-party partners. If you want to work more directly with your software vendor and their professional services team, you might consider Adobe, SDL, or one of the SaaS vendors in this report.

Don’t be surprised if you have to pay support fees to both Sitecore (for the software after year one, or if you want more than Bronze), and the integrator (for their customizations). To be fair, you can run into this with other systems as well, but it becomes more of an issue with Sitecore, because this vendor becomes a more distant player in any engagement, and this product often ends up being heavily customized in real-world implementations.

Finding experienced implementers is becoming less of a challenge, but it still requires a bit of legwork. Sitecore provides a progressive partner certification process, whereby partners are only required to maintain two certified Sitecore developers on staff. Sitecore also promises to launch a quality assurance program that requires partners to submit project assessments to Sitecore Customer Engagement Professionals (they don’t exist yet) for evaluations. There are four clients per one certified partner, which means that you must choose carefully here. Perform proper due diligence when selecting the right implementation partner.

On the plus side, the company maintains a lively public developer extranet for its partners and customers, featuring a blog, wikis, certified sample code modules, and active forums

(featuring thousands of posts). Just beware that this resource may prove hard to navigate and you'll be likely to spend quite a bit of time trying to find relevant and up-to-date information. Some developers express frustration that documentation is spotty, and while blog posts sometimes give great insights, they are sometimes only applicable to much earlier product versions and functions (that no longer exist). Don't expect Sitecore developer training to provide enough depth for your staff to understand how Sitecore works; the learning curve will be steep — and time consuming.

Sitecore bases its pricing on servers and concurrent CMS users. The full Engagement Platform Enterprise License starts at \$200,000. QA/Dev servers will cost you another \$60K each (you'll want to negotiate hard there). A user seat lists for about \$2,000.

Conclusion

Sitecore tries to push the limits of functionality and extensibility with a polished but complex offering, sold through channel partners who ultimately play a critical role in helping customers through a challenging setup and customization process. In choosing .NET as a development platform, Sitecore has been able to achieve tight integration with Microsoft technologies, while benefitting from a component-based architecture that, while more proprietary than Telerik or EPiServer, still plays well to third-party .NET libraries and extension modules.

Remember that Sitecore is not an “out-of-the-box” solution. Customers who want to be up and running with a production system within weeks (rather than months) need not apply. Unless you have considerable Microsoft-technology expertise in-house, you'll need outside help getting this system up and running. Your developers will find Sitecore more complicated (albeit more powerful) than Ektron, although it's a lot more consistent than the .NET/Java hodge-podge of SDL implementations.

Building its business around a channel partner ecosystem has proven a boon to Sitecore — though not always a boon to its customers. The fact that integrators tend to love Sitecore and push it hard among their customers as an omnibus solution should signal to you that it has evolved into a more complex platform than a productized offering. Moreover, the reliance on intermediaries puts an extra burden on you. You must now do *two* rounds of due diligence — one on the vendor, and one on the channel partner.

If you're an administrator or a contributor, much of what you think about this product will hinge on what you think of its polished and powerful (but potentially confusing and sluggish) user interface. We recommend that you do your own in-house human-factor testing to determine your various stakeholders' comfort levels with the product.

As a company, Sitecore continues on a rapid-growth trajectory, and maintains a very aggressive product-development roadmap. You will want to evaluate carefully whether usability is improving or suffering, as more functionality is added.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Mid-Range Platform Vendors: Roll-Up Comparisons

- Atex: Polopoly
- Drupal: Drupal
- EPiServer: EPiServer CMS
- Hippo B.V.: Hippo CMS
- Microsoft: SharePoint Server 2013
- Plone: Plone
- TYPO3: TYPO3

These offerings tend to serve mid-market companies, or departmental projects within larger enterprises. In particular, they target scenarios requiring above-average customization — often requiring an outside consultant. So it’s no surprise that this class of vendors tends to emphasize their communities, consulting partners and reseller channels.

This price/feature point appears to be quite attractive, and these vendors have been reasonably successful at selling into large companies. That doesn’t make them “enterprise” packages — despite what the vendor may tell you — but no matter; they are pretty good at providing a mix of content management and experience management capabilities, but you should note that you are trading off against potential customization and scalability limitations.

These packages typically carry initial price tags of US \$40–120,000+, but as always, the final fee can creep upward, depending on the final number of servers required or contributors in the system. Therefore, it pays to know up front if you are going to need clearly delineated development, staging, and production environments. Development and staging license fees are often negotiable. Nevertheless, product APIs almost certainly will not run as deep as those within upper-tier products, and definitely not as broad or available as the APIs you are likely to see among Upper-Range packages.

Speechheet Summary: Mid-range Platforms

	Atex Polopoly	Drupal	EPiServer
Geography	Global, with largest base in Sweden, UK, and Germany	Global	Europe, especially the Nordics, with a growing US presence
What's New	<ul style="list-style-type: none"> • New user interface • Performance improvements after migration to Solr • Improved content workflows and "My Desktop" overview • Text analytics integration with Temis engine • Tablet-specific publishing workflow and templates 	<ul style="list-style-type: none"> • Overhauled administrative interface and integrated content modeling • Slower performance than Drupal 6, but architectural improvements should make it easier to scale Drupal 7 • Expanded database support with Microsoft SQL Server & SQL Azure 	<ul style="list-style-type: none"> • Redesigned, cleaner user interface with drag-and-drop and in-page editing capabilities • MVT and A/B testing, campaign management, and other new digital experience features as part of the core CMS
Strengths	<ul style="list-style-type: none"> • Native content structures are very well suited to media sites in general and newspaper sites in particular • Combination of pages and placeless content makes it easy and intelligible to reuse content extensively • Users tend to like the straightforward and task-focused editorial interface • Enables power users to create new sections and populate them easily with content once the system is up and running 	<ul style="list-style-type: none"> • Designed explicitly for community-generated content, combining social interaction and web publishing into one platform • Very feature rich, with thousands of add-on modules • Nice facilities for ingesting and managing external feeds • Good support for "placeless" delivery and reuse of richly tagged content, as well as metadata-driven navigation • Highly configurable through the administrative user interface • Large and dedicated developer/user communities, with particular strength in Media industry • SaaS-based Drupal Gardens version may fit certain scenarios 	<ul style="list-style-type: none"> • Comparatively clean, Microsoft-based architecture, well suited to those customers with ample .NET developer resources • SharePoint integration will have strong appeal for customers who are already heavily invested in that platform • Company continues to exhibit strong technology vision and remains highly committed to latest Microsoft standards • Page/folder-oriented information model will feel familiar to many customers • Company focuses on core technology and leaves sales to resellers • Comparatively advanced digital marketing tools, including rules-based personalization, marketing dashboards, analytics, and ecommerce

(Continued)	Atex Polopoly	Drupal	EPiServer
Weaknesses	<ul style="list-style-type: none"> Leverages Java standards, but combines them to create an unusually proprietary system that may require substantial customization, which intensifies the risk of vendor lock-in Developers must have a solid grasp of the system's architecture when creating content types, workflows, and templates Back-end performance issues can be difficult to troubleshoot and prevent Consulting channel is unusually thin; customers outside of Sweden and Germany will have to rely on Atex professional services The critical lack of community and ecosystem around the product intensifies your dependence on Atex 	<ul style="list-style-type: none"> Drupal governance is informal and poorly defined outside of Acquia Despite the plethora of consultancies and SIs, truly experienced Drupal talent can be very hard to find — and keep You will likely need third-party modules of varying origin and reliability Limited versioning, clunky LDAP integration, and virtually nonexistent workflow capabilities make Drupal less suited for traditional enterprise environments Configuration management is highly problematic for all but small implementations; no support for proper environment separation Product is overkill for informational websites Highly trafficked sites may experience performance problems; there's a dearth of caching alternatives limit enterprise options here 	<ul style="list-style-type: none"> Licensing model can get costly in a hurry; numerous modules and packaging bundles are very confusing More of a development platform and less immediately "out of the box" than some other systems, requiring development in Visual Studio Busy and inconsistent editorial interface requires contributors to navigate many tabs and pop-up menus Power-user orientation might be overkill for some customers Page-centric approach may not work well for licensees looking for more component-based content reuse Only available via resellers, and they are comparably scarce beyond the UK and the Scandinavian home markets; growth in North America has been slow and uneven
Potential Fit	Multichannel Publishing, Ultra-Large Single Site	Community Oriented, Basic Digital Marketing	Microsites & Landing Pages, Basic Digital Marketing, Community Oriented sites
Unlikely Fit	Simpler or Mid-Range Scenarios	Global/Enterprise Digital Workplace, Global Enterprise, Multichannel Publishing	Informational, Digital Workplace
Compare To	Escenic, Magnolia, CoreMedia	Joomla!, Plone, TYPO3, DotNetNuke, Microsoft, WordPress	Ektron, Telerik, Kentico, Sitecore, SDL, Microsoft
Operating Systems	Linux, Windows; Java EE application server (JBoss, WebSphere)	Windows, Mac, Linux, UNIX	Microsoft Windows Server
Repository	JDBC-compliant database: Oracle, SQL Server	Databases: MySQL, MS SQL Server, SQL Azure, nominal support for PostgreSQL	Databases: Microsoft SQL Server, Oracle
Client	Any browser	Browser: IE, Firefox, Opera, Safari	Browser
App Platform	Java	PHP5	.NET
Licensing	\$100K and up	Open source — GPL v2	Server and site based; Enterprise license starts at \$17,900, with staging and testing tiers priced separately
Ownership	Privately held	N/A	Privately held; 220 employees, estimated \$42M in 2012 revenues

	Hippo	Microsoft
Geography	Primarily Europe, especially Benelux, with a growing footprint in North America	Global
What's New	<ul style="list-style-type: none"> • More advanced personalization based on personas • Better architectural separation of content and delivery tiers • New, more standard ("CK") rich text editor 	<ul style="list-style-type: none"> • SharePoint 2013 boasts major new enhancements and better pricing • Heavy-duty field experience with the new platform remains limited
Strengths	<ul style="list-style-type: none"> • Hippo B.V.'s technical and support acumen is highly regarded • Java platform built off contemporary standards • Can potentially support Multichannel Publishing with decent mobile support • Standards-based architecture is highly extensible in the right hands • Company is making an effort to expand in North America • Well-documented Java and REST APIs 	<ul style="list-style-type: none"> • Built on top of core SharePoint repository and library services, so you can leverage some SharePoint expertise for web publishing • Broad development platform to create content-oriented web apps • Microsoft has improved SharePoint's content contribution services and the system defaults to more standardized markup • Includes a full license to the powerful FAST Search engine • Platform boasts exceptionally broad, unofficial support community
Weaknesses	<ul style="list-style-type: none"> • Java platform built off contemporary standards • Deliberately optimized for Multichannel and Global Publishing • Standards-based architecture is highly extensible in the right hands • Company is making an effort to expand its partner channel • Strategy of integrating with third-party systems is suitable for enterprises pursuing a best-of-breed approach to digital marketing communications • Hippo B.V.'s technical and support acumen is highly regarded • New personalization features show strong promise 	<ul style="list-style-type: none"> • Requires enterprise web publishing teams to do things "SharePoint's way," with careful, up-front configuration planning • Very weak digital marketing features • Lack of packaged components means WCM efforts become development projects more quickly than collaboration scenarios • Surprisingly buggy and unexpected behavior, compared to more solid, previous releases • More onerous hardware requirements, comparatively pricey licensing, and high customization needs continue to make SharePoint 2013 a poor value for WCXM scenarios • Three-year upgrade schedule is a mismatch for fast-moving innovation cycles in this space
Potential Fit	Multichannel Publishing, Global Enterprise	Community-Oriented Site, SMB/Departmental Digital Workplace
Unlikely Fit	Global/Enterprise Digital Workplace, Advanced Marketing Portal	All Complex website scenarios
Compare To	Adobe, Magnolia, OpenCms, Drupal	Ektron, Sitecore, EPiServer, Telerik, Kentico
Operating Systems	Windows Server, a wide variety of Linux flavors	Microsoft Windows Server 2008
Repository	Apache Jackrabbit	Database: MS SQL Server 2008
Client	Browser: IE (8.0+), Firefox, Chrome, Safari	All major browsers supported
App Platform	Java: Tomcat, Oracle WebLogic Server	.NET

(Continued)	Hippo	Microsoft
Licensing	Community Edition available under Apache License 2.0; Enterprise Edition is sold commercially at a median license cost of \$50,000	Server and seats
Ownership	Privately held — Hippo B.V. holds all significant copyrights; 90 employees	Public (NASDAQ: MSFT)

	Plone	TYPO3
Geography	Global	Global, with largest base in Europe (Germany)
What's New	<ul style="list-style-type: none"> • New commenting system 	<ul style="list-style-type: none"> • Substantial UI modernization
Strengths	<ul style="list-style-type: none"> • Hybrid CMS + portal structure works well for website-in-a-box use cases • Strong support for XHTML and CSS, with strong accessibility bent for content contributors and consumers alike • Multiple authentication options • Resource compression for fast page transfers and bandwidth conservation • Natively generates friendly URLs • Highly tunable, policy-driven workflows • Support for clustering and "high-availability" architectures • Unusually broad set of multilingual content management tools • Large number of third-party add-on modules available • Unusually broad developer community with very mature governance structure, active forums, and a surfeit of online resources • Permissive GPL v2.0 license, with no plans to move to GPL 3.0 	<ul style="list-style-type: none"> • Large and comparatively active community keeps project dynamic • UI localized in an impressive forty-two languages • Exceptionally modular architecture eases customization, integration, and replacement of services • Wide array of optional plugins broadens feature set • Good workflow compared to most open source alternatives • Various clustering options for high performance • Runs unusually well on Windows ("WAMP")
Weaknesses	<ul style="list-style-type: none"> • Many moving parts, resulting in a steep learning curve for administrators, power users, and developers alike • No native support for any database other than "ZODB" • Customization requires significant Python expertise • Has no native API, and third-party modules vary hugely in quality • Coupled development, management, and delivery architecture can be problematic for managing large and/or busy sites • Version upgrades for Python or Zope (foundational technologies for Plone) can have unpredictable ripple effects throughout the CMS • Developer intercession is required to create and modify templates • Project momentum feels slower than competitors in past year 	<ul style="list-style-type: none"> • Out-of-the-box interface can be intimidating, especially for novices; very power-user oriented • Seems heavily reliant on optional modules of varying provenance, standards support, and accessibility • Developers must use obscure configuration language, "TypoScript," which has a steep learning curve • Templating requires a developer • Comparatively weak metadata support • Community-based development seems slow; version 5 has been in the works for more than three years • Community not as deep outside of Europe, particularly lacks breadth of system integrators in North America
Potential Fit	Informational Site, Microsites & Landing Pages, Community-Oriented Site	Informational Site, Community-Oriented Site, SMB/Departmental Digital Workplace

(Continued)	Plone	TYPO3
Unlikely Fit	Advanced Marketing Portal, Multichannel Publishing; Ultra-Large Single Site	Advanced Marketing Portal, Global Enterprise, Multichannel Publishing
Compare To	Drupal, eZ Systems, TYPO3	eZ Publish, Plone, OpenCms, Drupal
Operating Systems	Windows, Linux, UNIX, Mac	Linux, UNIX, Mac, Windows
Repository	Zope object store database (ZODB)	Any ODBC-compliant database: MySQL, Oracle, et al.
Client	Browser, WebDAV, FTP, External editor	Browser: Internet Explorer, Safari
App Platform	"Zope" application server	PHP 5.2+
Licensing	GPL 2.0	Open source: GPL
Ownership	N/A; "Plone Foundation" serves as project leadership directorate	TYPO3 Foundation

Category Summary: Mid-range Platforms

Phase / Attribute	Atex Polopoly	Drupal	EPiServer	Hippo	Microsoft	Plone	TYPO3
Technology							
Technical Administration & Security							
Threat Prevalence							
Authentication & Authorization							
System Reporting							
Multisite Management							
Cloud Services							
Development							
Configuration & Customization							
Integration & Extension							
Content Modeling							
Templating							
Performance							
Back-end Performance							
Site Caching & Delivery							
Content							
Contributor Experience							
Overall Usability							
UI Accessibility							
Contributing Content							
Authoring & Transformation							
Tagging & Taxonomy							
Content Reuse							
Media & Document Management							
Repository Services							
Content Lifecycle							
Workflow							
Globalization							
Archiving & Compliance							
Experience							
Publishing							
Standards Adherence							
Multichannel							
Mobile							

Phase / Attribute	Atex Polopoly	Drupal	EPiServer	Hippo	Microsoft	Plone	TYPO3
Digital Marketing							
Site & Campaign Analytics							
Testing & Optimization							
Segmentation & Personalization							
Social Media Integration							
Promotional Campaigns							
Community & UCG							
Workplace							
Collaboration & Networking							
Dashboard							
Ancillary							
Site Search							
Online Forms							
Module Ecosystem							
Vendor Intangibles							
Vendor Services							
Vendor Professional Services							
Channel Partner Services							
Support & Community							
Strategy & Roadmap							
Viability & Stability							

Key	
	Product does not provide this feature in any meaningful way.
	Product provides this feature, but is not as mature as its rivals
	Product provides this feature
	Product excels at this feature, <i>relative to other products in the same category</i>
	Product masters this feature, <i>relative to other products in the same category</i>

Scenario Fits Summary: Mid-range Platforms

Phase / Attribute	Atex Polopoly	Drupal	EPiServer	Hippo	Microsoft	Plone	TYPO3
Simpler Site							
Informational							
Microsites & Landing							
Mid-Range							
Basic Digital Marketing							
Mobile Site							
Community Oriented							
Complex Site							
Advanced Marketing Portal							
Global Enterprise							
Multichannel Publishing							
Ultra-Large Single							
Digital Workplace							
SMB/Departmental							
Global/Enterprise							

Key	
	Product does not provide this feature in any meaningful way.
	Product provides this feature, but is not as mature as its rivals
	Product provides this feature
	Product excels at this feature, <i>relative to other products in the same category</i>
	Product masters this feature, <i>relative to other products in the same category</i>

Atex: Polopoly

atex.com

Product at a Glance

Specsheet	Atex: Polopoly 10 Summary
Geography	Global, with largest base in Sweden, UK, and Germany
What's New	<ul style="list-style-type: none"> • New user interface • Performance improvements after migration to Solr • Improved content workflows and “My Desktop” overview • Text analytics integration with Temis engine • Tablet-specific publishing workflow and templates
Strengths	<ul style="list-style-type: none"> • Native content structures are very well suited to media sites in general and newspaper sites in particular • Combination of pages and placeless content makes it easy and intelligible to reuse content extensively • Users tend to like the straightforward and task-focused editorial interface • Enables power users to create new sections and populate them easily with content once the system is up and running
Weaknesses	<ul style="list-style-type: none"> • Leverages Java standards, but combines them to create an unusually proprietary system that may require substantial customization, which intensifies the risk of vendor lock-in • Developers must have a solid grasp of the system’s architecture when creating content types, workflows, and templates • Back-end performance issues can be difficult to troubleshoot and prevent • Consulting channel is unusually thin; customers outside of Sweden and Germany will have to rely on Atex professional services • The critical lack of community and ecosystem around the product intensifies your dependence on Atex
Potential Fit	Multichannel Publishing, Ultra-Large Single Site
Unlikely Fit	Simpler or Mid-Range Scenarios
Compare To	Escenic, Magnolia, CoreMedia
Operating Systems	Linux, Windows; Java EE application server (JBoss, WebSphere)
Repository	JDBC-compliant database: Oracle, SQL Server
Client	Any browser
App Platform	Java
Licensing	\$100K and up
Ownership	Privately held

Summary

Atex Polopoly is a Java-based system that revolves around a mix of pages and attached articles, which become reusable in various “publishing queues” elsewhere. This combination renders the system as well suited to media sites, and may work well if employed for similar site structures in other industries.

However, while the system employs various Java standards and popular components such as Apache Solr and Velocity, it combines it with a proprietary architecture. This, and quirks like Polopoly’s own JavaServer Faces alternative (called Orchid) may be documented, but aren’t broadly supported by any community. After an initial implementation by Atex professional services, it can be a challenge to maintain and customize the system yourself.

Consider Polopoly for (print) media websites, especially in Northern Europe, and be sure to compare it to Escenic.

Scenario Fits	
Simpler Site	
Informational	<input type="radio"/>
Microsites & Landing	<input type="radio"/>
Mid-Range	
Basic Digital Marketing	<input type="radio"/>
Mobile Site	<input type="radio"/>
Community Oriented	<input type="radio"/>
Complex Site	
Advanced Marketing Portal	<input type="radio"/>
Global Enterprise	<input type="radio"/>
Multichannel Publishing	<input type="radio"/>
Ultra-Large Single	<input type="radio"/>
Digital Workplace	
SMB/Departmental	<input type="radio"/>
Global/Enterprise	<input type="radio"/>

Introduction

When Sweden’s largest newspaper Dagens Nyheter needed a website in 1996, a group of Java developers set out to create a content management system, which became the Polopoly CMS. In 2008, the Swedish company was acquired by the UK company, Atex. Founded in 1973, Atex had been an early innovator in editorial content management, mostly for print media. While the vendor had managed to market successfully around the globe, growth fell behind as market demand for print automation slowed down. In 2009, the company was reorganized to integrate Polopoly, with digital media as its major focus going forward.

Polopoly development has been relatively slow and it is not unusual for Atex to be a couple of years late with promised release dates; the software has undergone steady evolution rather than enduring any major overhauls. To be fair, Atex has sometimes introduced new functionality in minor versions (like the SiteEngine), which is something that other vendors would have released as a major version number for marketing purposes.

Technology

Technical Administration and Security

Polopoly is a Java EE system that leverages several Java standards, such as Java servlets, EJB, JSP, JDBC, and JMS. However, the architecture itself is quite proprietary. For instance, instead of JavaServer Faces (JSF), Polopoly uses its own application framework called Orchid.

Central in the architecture is the CM Server (an EJB application). It requires a J2EE 1.4 application server to run (JBoss seems to be the most common choice, but WebSphere has also been tested by Atex).

Content delivery is dynamic and coupled; the delivery is handled by the “Front Web

Application,” a web application run in a web application container (such as Tomcat or WebSphere). Likewise, the editorial and administrative UI is a web application communicating with the CM Server.

Users can be managed by Polopoly, or you can add an external LDAP provider. You can assign rights to individuals or groups; rights are inherited, but also can be overruled. The mechanism isn’t particularly sophisticated, but will allow you to lock-down customized functionality for most scenarios.

Content is modeled using an “input template,” which defines the fields. The content type also has a “policy,” which defines how its data is accessed, as well as an “editor” and “viewer” element so it can be accessed in the editorial UI. Similarly, there is an “output template,” which defines what is output, and which JSP to use to render it within the front-end controller. These (and many other configurations) are defined in XML, and then imported into the system. As you can imagine, defining content types and presentation templates in Polopoly is really a developer’s job — and developers need a good understanding of Polopoly’s quirks, including the Orchid framework.

Development

The basic elements of developing within Polopoly can be broken down as follows:

- **Java** - Uses the aforementioned Orchid framework to build controllers
- **XML** - Builds “input templates” that define content modeling rules
- **Apache Velocity and/or JQuery** - Creates the “output template” that defines how the content is displayed

The Polopoly “SiteEngine” enables editors to create page layout, using pre-defined layouts and layout elements. New pages can be created with a form, where elements are added to the columns in the page. For developers, the advantage is that Velocity can be used for templating (reducing the amount of code required). As with other page editors, the drawback is that the increased editorial flexibility comes at the price of more up-front development.

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

Most of Polopoly’s components can be scaled. For instance, for better delivery performance, multiple Front Web Applications can be employed with a caching server and load balancer in front of them.

However, the CM Server can only be scaled up, which — unless a passive failover is also provisioned within your architecture — creates a possible single point of failure, since the rest of the infrastructure revolves around this essential component. Atex says this has never happened, and that even a catastrophic failure of the CM Server wouldn’t affect website presentation immediately, since the Front Web Applications run autonomously. Atex also says that in order to achieve optimal system resilience, it recommends two back-end installations: a primary and a secondary. If the primary instance fails, you can activate a fallback mechanism (manually or automatically) to switch over to the secondary installation. However, the process is not straightforward, and you need to ensure that the back-end’s multiple Java processes are all safely switched at the same time to maintain a controlled state. The cost of adding resilience to Polopoly, therefore, means additional hardware and maintenance investments.

Performance

In the past, there were frequent reports of performance problems with the editorial interface, but, Polopoly finally integrated the well-regarded Solr in version 10 to act as the sole back-office repository search. This seems to have removed prior bottlenecks with performance, but you should test this new search functionality carefully with your own content.

Polopoly relies on a front-end Model View Controller (MVC) to build pages dynamically upon request (with caching to assist). As with any instance using this delivery method, give careful consideration to scaling this tier of the architecture. Given the proprietary nature of Polopoly and the paucity of community support, assistance with this is likely to fall back on the best practices suggested by the vendor and its professional services team.

Content

Contributor Experience

Polopoly’s user interface has undergone a major overhaul in version 10, following many usability complaints from users on version 9 (which first came out in 2005).

Now with a more contemporary look, the basic elements of the interface have remained more or less in place, albeit arranged around a “My Desktop” paradigm. Using My Desktop, you can get quick access to tasks in progress, such as those generated by workflows.

The tabbed approach for accessing elements of functions within a content item (e.g., preview) remains, so if you’re familiar with the version 9, there is no change. For new users, working with the interface is as much about understanding the structure of the eventual publication, as it is locating the functionality. With virtually all

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

Polopoly usage remaining within the print media industry, this publication focus is natural — and perhaps sensible. Outside of the publishing vertical, this may be a limitation.

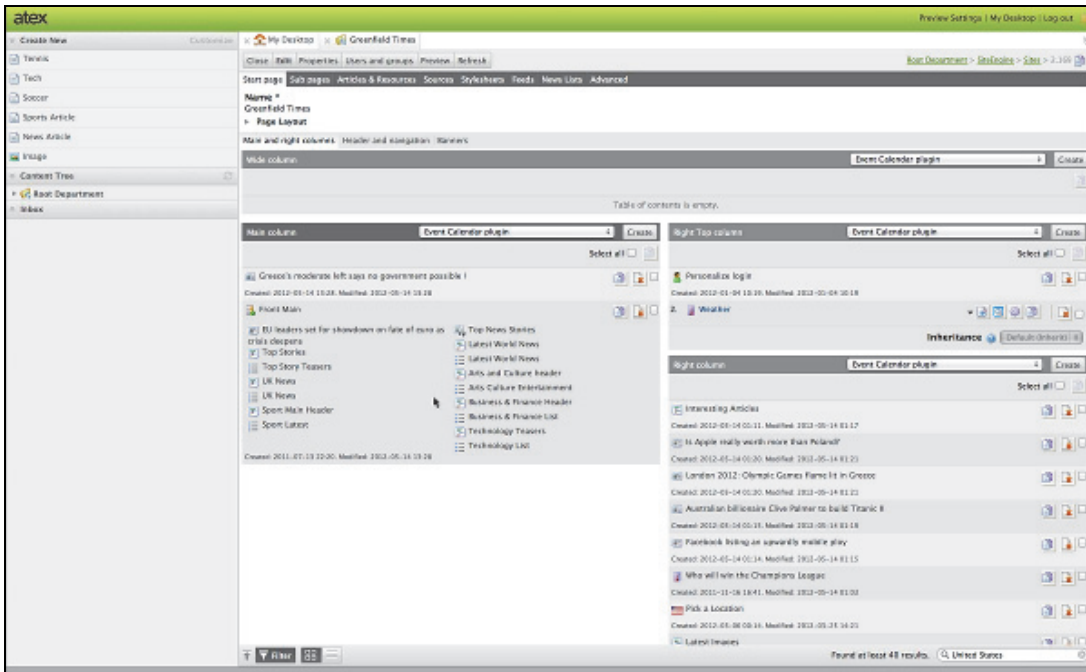


Figure 174. Polopoly’s contributor interface for page creation.

Despite the shine of version 10, the interface is still only available in only two languages — English and Swedish — but in theory, you could localize it yourself. The vendor says regional professional services teams supply other languages (such as German, Vietnamese, Chinese, Spanish, Portuguese, and Italian), but how these are supported in practice — say for future upgrades — is unclear.

Contributing Content

Polopoly sites consist of index pages, which can have similar child index pages underneath them. You go to the parent page, and from there create a new “subpage,” which brings up the forms-based designer (the contents of which are defined by “input templates”). The actual “articles” also belong to a “page,” and are displayed through the page’s “publishing queue.” Think of each page as a layout with several columns (laid-out explicitly at creation), and one displays a list of titles and teasers, linking to individual article pages (which are automatically generated).

Note that Polopoly is designed primarily to assemble content, not to originate it. Feeds from a print editorial system — Atex’s primary business is in supplying this — augmented by those from news syndication sources (such as Reuters or Associated Press) are typically the bulk of what flows through a Polopoly CMS instance. The interface reflects this, allowing you to assemble pages from these feeds — referred to as “publishing queues” — and to place formatting objects (such as an events calendar). The queues can be generated both manually and automatically.

The result is very much what you would expect from a newspaper website. The Greenfield Times (Polopoly’s demo website, a version of which is supplied out-of-the-box as a development starter) has “Sports,” “News,” and “Travel” sub-pages. Each of these has queues of its own articles. However, Polopoly’s real strength is in reuse. Articles can also show up in other publishing queues, automatically via (metadata) criteria, or a manual selection. This can be difficult to achieve in other systems, but with Polopoly, it’s out-of-the-box functionality. As a result, the CMS works particularly well for news sites, which tend to follow the kind of site-tree structure that Polopoly was designed to do.

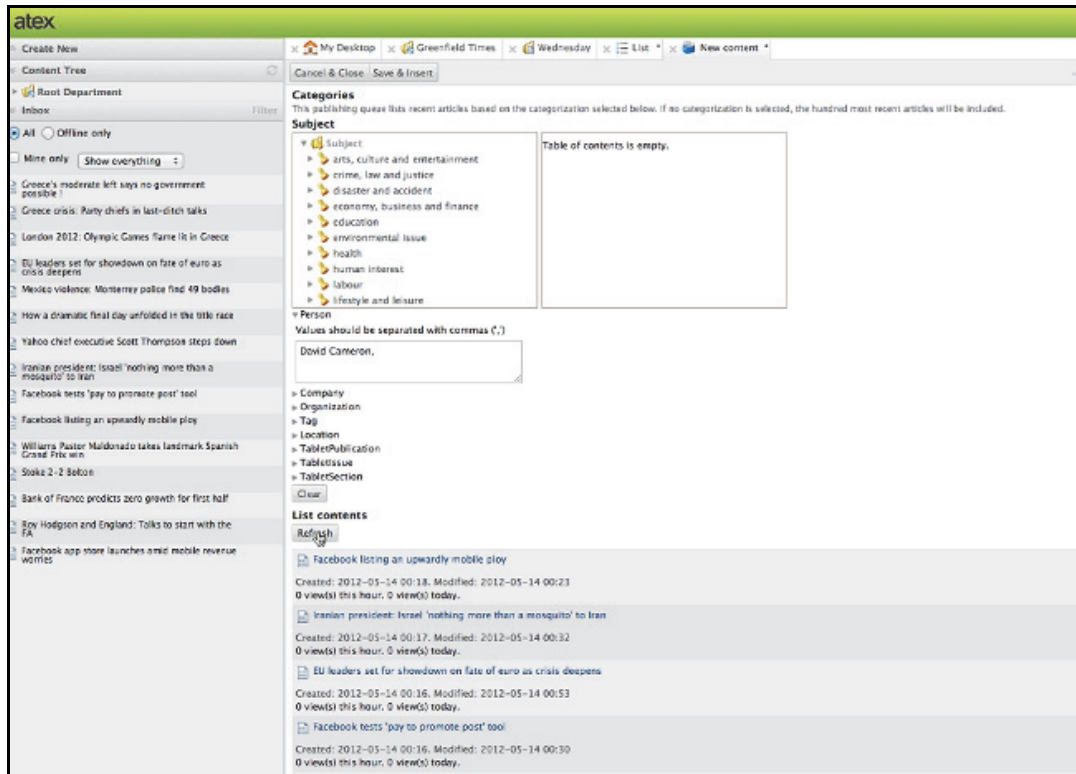


Figure 175. Publishing queues can be driven by metadata.

Atex maintains that it would work equally well for any kind of other website, and if you’re planning for a similar structure with section indexes and placeless content, they may be right. For instance, it has appealed to several universities that found the layered structure to be a good match for their decentralized organization. However, in practice, most customers reside in (print) media. This is not just because Atex is firmly rooted there; Polopoly has a strong presupposition of a site structure, which is a particular mix of pages and placeless content. If that doesn’t agree with you, you’ll find yourself struggling to bend the system’s rules.

For rich text editing, Polopoly uses the popular FCKeditor with most options turned off in its sparse configuration. Articles are created off the index pages. Article content is structured; as with Escenic, images aren’t added in the rich text field, but they are picked separately. Polopoly has a Java applet that enables editors to do simple image manipulation, like cropping and rotating — but not changing colors or contrasts.

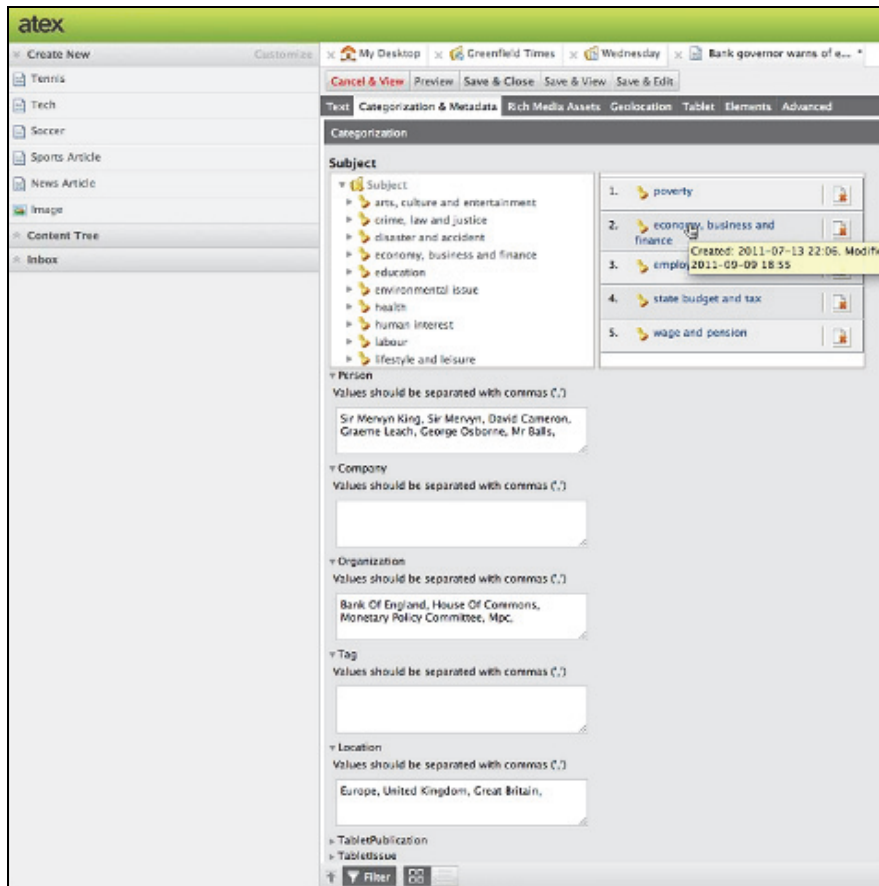


Figure 176. Integration with an optional text mining engine can aid metadata management for queues.

You edit categorization and other metadata from the same pane as the main content — something that editors would love to see in other products that tend to hide these three tabs. With this metadata-driven queue method, Polopoly has an integration with a text mining tool from Temis (available for an additional licensing fee). Temis will auto-suggest tags for each content item, potentially enabling you to create fine-grained, topic-based publishing queues. If you are considering this optional add-on, it is important to do detailed testing, as text mining engines often require considerable customization and fine-tuning to produce accurate and relevant results. In addition, text mining engines are highly language dependent.

Content Lifecycle

The system can keep versions of articles, but there is no visual-compare feature. Workflow is based on stages; the content is tagged with the stage it is in as it changes from one state to another. This kind of linear workflow works well in a standard editorial process, but is ill suited for complex branching workflows that are required for translation or compliance. Note that there is no workflow editor; creating a workflow is customized development work.

Experience

Publishing

Polopoly produces relatively clean code, and it should be easy to create a standards- and accessibility-compliant site. The system uses intelligible, human-readable URLs, which follow the format:

`cmlink/sectionname/articletitle.`

Unfortunately, it also adds a unique identifier to each article in the form of a dotted number; this doesn't break any contemporary webmaster conventions, but it is inelegant.

With the launch of version 10, Apex also released a new tablet publishing system. In essence, this is a specific set of workflows, as well as HTML5 and CSS3 templates designed to keep a tablet version of your publication in sync with the website itself. As a result, you can apply the same publishing-queue methods employed elsewhere within the system to your tablet. This add-on is not designed to produce operating system-specific builds (such as apps for iOS or Android) but rather it is a responsive design to provide a browser experience that is properly graded for the consuming device. As with all version 1.0 products, a little caution — and plenty of testing — are well advised.

Digital Marketing

Don't look for much here. To be sure, the CMS comes with several built-in modules such as subscriptions, polls, commenting, and a forum. These are not particularly sophisticated compared to best-of-breed, third-party products, but they get the job done. Unlike many add-ons in competitor's products, Polopoly has actually added spam control (using WordPress's Akismet), naughty filtering, and CAPTCHAs. However, while all of the API hooks and infrastructure are there, you won't stray too far from what's available out of the box without some serious development. If you want to build a more "social" website, consider Drupal or EPiServer, instead.

The system's Statistics Server hooks into the Orchid framework to tally views. This means it can be used for application profiling (to diagnose performance issues), and personalization features, although they are far from out of the box.

Ancillary Services

Polopoly employs Solr for content search.

There is really no module ecosystem; modules are developed by Atex itself.

Experience	
Publishing	
Standards Adherence	<input type="checkbox"/>
Multichannel	<input type="checkbox"/>
Mobile	<input type="checkbox"/>
Digital Marketing	
Site & Campaign Analytics	<input type="checkbox"/>
Testing & Optimization	<input type="checkbox"/>
Segmentation & Personalization	<input type="checkbox"/>
Social Media Integration	<input type="checkbox"/>
Promotional Campaigns	<input type="checkbox"/>
Community & UGC	<input type="checkbox"/>
Workplace	
Collaboration & Networking	<input type="checkbox"/>
Dashboard	<input type="checkbox"/>
Ancillary	
Site Search	<input type="checkbox"/>
Online Forms	<input type="checkbox"/>
Module Ecosystem	<input type="checkbox"/>

Vendor Intangibles

Atex is majority owned by Kistefos, a Norwegian investment firm (solely owned by Christen Sveaas). Atex currently has about 600 employees, and says it serves 850 customers in 40 countries, although there is no clear breakdown as to which part of this is print media, and which part is web content management.

There is a lively services ecosystem around Polopoly in Sweden, and there are several integrators in Germany that specialize in implementing the solution. However, in most parts of the world, Atex would not only be your vendor, but also your integrator, with Atex professional services there to get you up and running. Because of the difficulty of finding knowledgeable resources elsewhere, you'll be mostly on your own after that.

Atex offers various courses for Polopoly, but the calendar is restricted mostly for basic courses to be held in Stockholm. Your training will have to be mostly on-the-job; make sure to be well acquainted before Atex leaves. The technical documentation is quite good, but unlikely to be sufficient to implement the complex system yourself.

There is no coherent customer community around Polopoly. Atex organizes some local user meetings, but it is haphazard and infrequent. There is a public forum for Polopoly (run by two users), which sees some activity, but it is primarily in Swedish and updates are infrequent.

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

Conclusion

Polopoly is one of the few systems that has found a pragmatic way to cope with the challenges of reusing placeless content. It offers editors more control than tag-based delivery (as with Drupal) by using pages as hooks for sections. Although it previously looked somewhat staid, the editorial interface often was preferred over Escenic's rich client, which is too rich for some users. The new interface shouldn't put off existing users, and it adds a bit of polish to attract new ones. Moreover, Polopoly's assumption of what site structure should be — and how it should be populated — could match the goals of what many media companies want to implement.

However, the system is hard to master, and you may overly rely on the vendor's support and professional services; keeping it running smoothly is not easy. More importantly, make sure that the Atex web publishing paradigms match up with what you need. If you bend this system to something it's not, it will come at a price — and it might break, as well.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Drupal: Drupal

drupal.org

Project at a Glance

Specsheet	Drupal: Drupal 7.12 Summary
Geography	Global
What's New	<ul style="list-style-type: none"> • Overhauled administrative interface and integrated content modeling • Slower performance than Drupal 6, but architectural improvements should make it easier to scale Drupal 7 • Expanded database support with Microsoft SQL Server & SQL Azure
Strengths	<ul style="list-style-type: none"> • Designed explicitly for community-generated content, combining social interaction and web publishing into one platform • Very feature rich, with thousands of add-on modules • Nice facilities for ingesting and managing external feeds • Good support for “placeless” delivery and reuse of richly tagged content, as well as metadata-driven navigation • Highly configurable through the administrative user interface • Large and dedicated developer/user communities, with particular strength in Media industry • SaaS-based Drupal Gardens version may fit certain scenarios
Weaknesses	<ul style="list-style-type: none"> • Drupal governance is informal and poorly defined outside of Acquia • Despite the plethora of consultancies and SIs, truly experienced Drupal talent can be very hard to find — and keep • You will likely need third-party modules of varying origin and reliability • Limited versioning, clunky LDAP integration, and virtually nonexistent workflow capabilities make Drupal less suited for traditional enterprise environments • Configuration management is highly problematic for all but small implementations; no support for proper environment separation • Product is overkill for informational websites • Highly trafficked sites may experience performance problems; there's a dearth of caching alternatives limit enterprise options here
Potential Fit	Community Oriented, Basic Digital Marketing
Unlikely Fit	Global/Enterprise Digital Workplace, Global Enterprise, Multichannel Publishing
Compare To	Joomla!, Plone, TYPO3, DotNetNuke, Microsoft, WordPress
Operating Systems	Windows, Mac, Linux, UNIX
Repository	Databases: MySQL, MS SQL Server, SQL Azure, nominal support for PostgreSQL
Client	Browser: IE, Firefox, Opera, Safari
App Platform	PHP5
Licensing	Open source — GPL v2

Specsheet	Drupal: Drupal 7.12 Summary
Ownership	N/A

Summary

Similar to DotNetNuke, the Drupal project approaches content management from a different angle than traditional Web CMS tools constructed primarily as web publishing tools. Drupal’s origin in community collaboration blurs the line between contributor and audience — with blogging, commenting, tagging, and voting functionality. It is as much a social software platform as it is a web content management system.

While competing packages have also added these features, in Drupal they are bred in the bone. Still, Drupal is deployed for a wide range of basic content management purposes, from corporate brochure sites, to online storefronts, to news sites, and to .gov sites. In many ways, it’s similar to WordPress — although

WordPress remains somewhat more of a blogging platform than a full-blown WCM.

Note that Drupal is not a simple platform, especially given the array of potentially necessary third-party modules you may need to install. A commercial firm, Acquia — led by Drupal founder Dries Buytaert — has sprung up to offer services and support, particularly for high-traffic environments.

Drupal tends to be deployed most frequently for community-oriented applications. While it is also frequently used for simple website hosting, managing static websites does not fully leverage Drupal’s unique capabilities, and may expose weaknesses in more conventional WCM functionality. For all its hype, Drupal continues to lack in key enterprise services.

In short, Drupal has made some waves for itself, but has also been abandoned by some early adopters subjected to its limitations. With respect to Drupal’s PHP-based competitors (simpler systems such as WordPress and Joomla), Drupal is the most functional of the batch, but it still has some severe shortcomings with complex deployments. Scaling Drupal is not trivial.

Nevertheless, the core concern for the majority in the highly distributed Drupal community still revolves around Web 2.0-style collaborative functionality over simpler web publishing, and larger enterprises turn to Acquia to try to solve scalability issues.

Perhaps the biggest challenge you will face is finding experienced Drupal talent. There’s a chronic misimpression that an everyday PHP developer can quickly pick up Drupal; it’s actually harder than that, and given the platform’s popularity, you will find a global shortage of truly experienced Drupal specialists.

Scenario Fits	
Simpler Site	
Informational	<input type="checkbox"/>
Microsites & Landing	<input type="checkbox"/>
Mid-Range	
Basic Digital Marketing	<input type="checkbox"/>
Mobile Site	<input type="checkbox"/>
Community Oriented	<input checked="" type="checkbox"/>
Complex Site	
Advanced Marketing Portal	<input type="checkbox"/>
Global Enterprise	<input type="checkbox"/>
Multichannel Publishing	<input type="checkbox"/>
Ultra-Large Single	<input type="checkbox"/>
Digital Workplace	
SMB/Departmental	<input type="checkbox"/>
Global/Enterprise	<input type="checkbox"/>

Introduction

In 2000, when project founder Dries Buytaert first conceived of Drupal as a student at the University of Antwerp, he was envisioning a bulletin board-like system where a community could leave messages for each other. In January 2001, the source code for the software behind his nascent site was released under the GPL to enable others to experiment and innovate with the platform. Since that time, the Drupal community has been rapidly expanding — using and promoting Drupal with the slogan “Community Plumbing.”

Perhaps Drupal’s biggest break came in late 2009, when the White House site was re-launched (with help from Acquia to advise on scaling the system). At the same time, another White House initiative (called *recovery.gov* — originally quickly launched on Drupal) switched to SharePoint, which was a better match for the data mashups required in that site.

Mainstream enterprises have been starting to pay attention to Drupal as well, particularly traditional media companies looking for ways to break into community-generated content. Large web properties like MTV UK and Spread Firefox are deployed on Drupal. However, the popular satirical news site, The Onion (which for years ran on a heavily modified Drupal 4 implementation) decided to abandon the software. Reportedly, the customizations that were required to scale the system to their needs became so labor intensive, it was easier to continue on a completely custom platform rather than to try to upgrade to a newer version of Drupal.

Acquia, Inc., a venture-funded company, was created in 2008 with the Drupal founder, Dries Buytaert, as CTO. Originally, the company set out to branch off a “hardened,” certified Drupal version as “Acquia Drupal,” in the same vein that for instance that DNN Corp is offering a guaranteed DotNetNuke version. More recently, the company started to place more emphasis on services, consulting, and training. However, it doesn’t build the actual implementations itself (partners do this, with expert help from Acquia). A large chunk of Acquia’s work is fixing agency-built Drupal sites that don’t turn out to handle high traffic very well. If you plan to implement Drupal for millions of visitors (rather than thousands), get Acquia involved from the beginning, and not as an afterthought; few other consultancies have comparable experience here. Of course, you’ll have to pay for this.

Acquia also offers the “Acquia Network,” which is mostly a direct support line, and it includes the “Acquia Network Module,” with an update notification service, and a heartbeat monitor for your site.

An increasing part of Acquia’s business is offering Drupal as SaaS, using Amazon’s platform. The company says that for a recent merger in the airline industry, it quickly set up an Amazon environment to scale Drupal to handle hundreds of millions of visits (and then scale back to a much more modest capacity after the first few weeks). This, of course, is exactly the kind of scenario where, if you want to use Drupal, you should talk to Acquia. The alternative would be to use the Pressflow distribution, which has modifications for high traffic scenarios — in fact, quite a few of the scalability improvements in Drupal 7 came from Pressflow 6. Be warned, however, that running Pressflow requires a lot of specialist knowledge.

Version 7 was released in January 2011, reaching version 7.12 in February 2012. There was only one year between the version 5 branch and the release of version 6 in February 2008. Version 7 took considerably longer to complete and was pushed back several times before finally being released.

Drupal 7 addresses several of Drupal’s known deficiencies — in particular, to make Drupal more user friendly. Acquia sponsored a team that completely overhauled the administrative

interface. Version 7 also incorporates “custom fields” in the Drupal core (in version 6, this required the add-on module CCK). Essentially, custom fields allow you to define your own content types. This is a basic capability of most systems in our WCM research, but for those coming from a social software background, it’s a new trend.

Drupal Gardens (Acquia’s commercial SaaS solution, based upon version 7) was formally released in March 2011 after a year in open beta. Taking their lead from what WordPress has done in creating a commercial hosted offering spanned off their open source offering, Drupal Gardens is a similarly tiered system. Starting from a low entry point, prices increase based upon usage (users, storage, and bandwidth), functionality (analytics, SEO) and support (standard and priority support ticket offering).

March 2012 saw the advanced announcement of “Enterprise Drupal Gardens,” a further venture into commercialization from Acquia. As with the Gardens’ flavor of Drupal, a dearth of enterprise features make this offering less suitable for traditional enterprise environments. Review these features carefully before investing.

Technology

Technical Administration and Security

Drupal is built on the popular LAMP (Linux, Apache, MySQL, and PHP) platform, although there are many implementations running on Windows and other operating systems. For the database, MySQL is recommended. Drupal 7 requires PHP5 and MySQL 5. MS SQL Server and its Azure sibling are now supported, but not widely used within the community. PostgreSQL (version 8 and up for Drupal 7) is nominally supported, but Drupal does not support Oracle. Although Apache is the recommended web server, Microsoft IIS can also be used if PHP has been configured on it.

Drupal’s technical design is based on a principle of having a slim core that is extensible by adding custom modules. The development of Drupal has kept true to this goal and the core continues to remain lean, while there are literally thousands of modules that perform specific functions. Several primary modules are packaged in the core distribution. It is typical for a Drupal implementation team to utter the words, “There is a module that does that.”

However, it takes time for the module developers to catch up to the architectural changes introduced by each major release of the core. Module developers are responsible for testing and giving honest descriptions on the state of their modules (such as alpha, beta, and what version of the core the module with which is compatible). Because development of Drupal 7 was drawn out, many major modules should be available from day one. However, many implementations depend on more obscure modules that can perform essential functions: for instance, there is no “link picker” available – or even in development – for Drupal 7 yet. Be careful to find out whether you’ll actually be able to implement what you need.

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

The Drupal website has recently been overhauled to make it easier to find modules and see how the community rates them. Even popular modules don't get a lot of ratings yet – the site helpfully displays how many Drupal implementations report back that they use a module, but only a handful will actually post feedback. In many cases, there are dozens of closely resembling alternatives, but it'll be difficult to measure. Unfortunately, the best information about the quality of a module is still usually word of mouth by Drupal specialists, who are constantly experimenting with new modules as they are released. *Caveat emptor*.

Installing a module involves uploading a package of files into the “modules” directory and enabling it through the administrative UI. In Drupal 7, it's also possible to upload the modules through the interface, so FTP access is no longer required. The module management lists dependencies on other modules, and will refuse to enable a module if these aren't met. The process is slowly getting easier, but Drupal is several steps behind TYPO3 – where extensions can be searched from the interface and installing them, including dependencies, is point-and-click.

Configuration management can become a nightmare when dealing with multiple environments such as development-staging-production, or multiple development environments. This might be acceptable for small sites with a one- or two-person development team (which historically has been the case with a majority of Drupal sites), but it is virtually unmanageable for larger-scale operations.

Drupal is a tightly coupled system that publishes content dynamically. Drupal's presentation tier (which is the same as the content production environment) is portal-like, with elements (called “blocks”) placed on regions of a page. Blocks are used to surface information from various features installed within the application. For example, the events module ships with an event calendar block that shows upcoming events. For each block, an administrator is able to control where the block will appear by assigning it to a region, the conditions under which it should appear with a PHP “if” statement, and whether a user can select to hide the block. Care should be taken here, because there is no safe way to “test” the code.

The administration interface is built into your outward-facing Drupal website. Some enterprises may cringe at the security implications of this design, given that it offers such powerful controls, which become accessible outside of the firewall. This openness has exposed a large number of security vulnerabilities over the history of the Drupal project. Once logged in, a user with administrative privileges will have a menu bar on top of the site, including a “configuration” item.

Access Control requires a combination of permissions, roles, and users. User Authentication can be done by Drupal locally or by external sources such as OpenID. LDAP integration is provided through an add-on module.

Drupal provides basic logging and reporting. All the events are captured in the log and then reports are generated. The default install includes some report pages that show “top access denied pages,” “top page not found errors,” and “top search phrases.” The reports are simple lists with the ability to neither filter nor create ad hoc queries. There is also a more generic “event log” that shows all logged events, which can be filtered, but by default, only system messages and errors are logged; there is no user auditing.

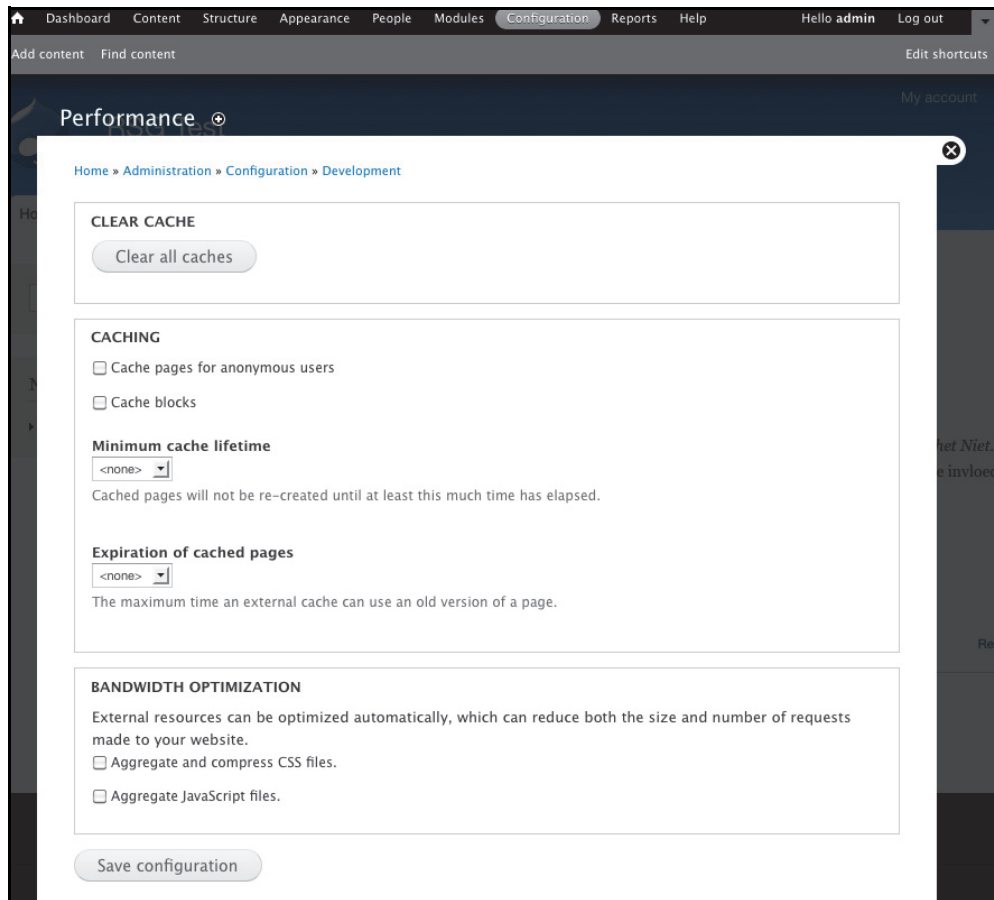


Figure 177. Configuring Drupal 7 through the administrative interface. Note the basic caching and bandwidth optimization features that have been added.

Acquia Drupal, through the “Acquia Network,” provides several hosted services to make monitoring your Drupal site easier. You can get notifications if your site is unreachable; unlike some third-party services, this isn’t done by the service checking to see if your site will respond, but by Drupal itself pushing a heartbeat to Acquia. This also means that your site doesn’t necessarily have to be open to the public for this to work, since the communication with Acquia can be secured by using SSL. Note that this service requires a paid subscription.

Development

PHP developers typically find Drupal easy to learn, because Drupal customizations are written in standard, straightforward PHP and SQL, rather than a proprietary scripting syntax or elaborate indirection. The key concept to grasp is Drupal’s “hooks” framework, which enables the Drupal core to call functions on installed modules. Once this concept is mastered, code is relatively easy to read, but script files are very long, so developers will want a good IDE that can navigate the code by function. In Drupal 6, database queries were written directly in SQL; in Drupal 7, there is a database abstraction layer, but it is still direct.

In Drupal 6, every item of content was called a Node. Because every content asset is an instance of this abstract class “Node,” a Drupal developer is able to do many object-oriented types of things with content assets, such as create collections of different types of content and access them in uniform ways.

In Drupal 7, new content types can be created from the “structure” menu item in the UI, but that’s only half the story. The new version has further abstracted the way content is treated, and the top-level item is now an “entity,” and nodes are just one of the entity types.

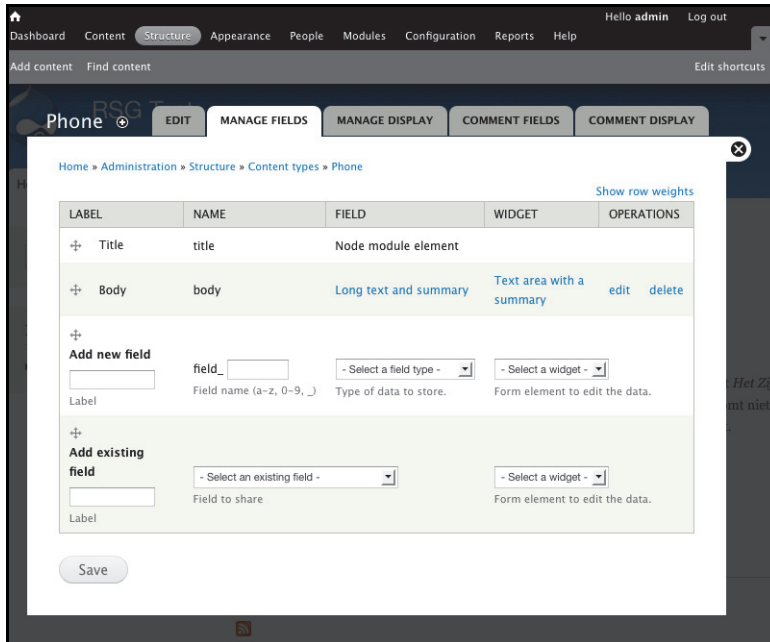


Figure 178. Creating a new content type in Drupal 7.

The layout and branding of a Drupal site is controlled by packages of templates, style sheets, images, and JavaScripts called “themes.” Multiple themes can be installed on a Drupal site and, unless prevented by the administrators, users can select from a set of enabled themes.

A theme defines regions to which an administrator can assign zero or more blocks and menus. For consistency's sake, there are several standard regions: header, left, right, and footer. This enables themes to be more swappable. However, heavily customized Drupal sites will have custom regions to increase the level of control over the layout. The easiest way to create a custom theme is to download one of the themes available from the Drupal website and modify it.

In Drupal 7, the PHPTemplate theme engine has become the default. Like the content model, it retains some of the familiar concepts of the previous versions. However, in Drupal 7, all content is initially passed to the template as one array. This means that there is now a much cleaner and more uniform way to deal with granular content, which with Drupal 6 and CCK required long lines of code, created performance issues, and possible security risks.

Templating is still somewhat peculiar and Drupal specific. In version 7 however, it's more flexible, but more complicated than WordPress' relatively direct function calls. At the same time, it's a lot easier to grasp than TYPO3's TypoScript.

A nice bonus of Drupal 7's templating is the integration of RDFa functionality. Apart from page metadata, this can also markup the fields of a node in the output semantically. Google is

adding support for extracting “rich snippets” such as reviews, people, and events. If these are available in your pages in RDFa, the search engine will automatically pick them up and display them in the search results appropriately.

Performance

In Drupal 6, because module and block developers could write their own SQL queries and administrators could place any number of blocks on a page, a single page load could execute hundreds of individual queries. Obviously, that's programmatically intensive and can really slow things down. Drupal 7 has abstracted and standardized the content model, database access, and templating, which should help improve this. In practice, however, the added abstractions increase the load on the server, and Drupal 7 is somewhat slower. The development team has conceded this to be true, but says that version 7 should be easier to scale.

However, scaling a Drupal implementation to very high-bandwidth scenarios is still not trivial. In the past few years, a number of intensively used websites have started using the software, so there is more experience in how to deal with this. The Pressflow distribution has made some modifications to run Drupal more efficiently, adding Varnish, and swapping Drupal's built-in search for Apache Lucene for better performance. Some of the enhancements Pressflow made to Drupal 6's core have been integrated into Drupal 7. However, in reality, you're still likely to require input from Acquia if you want to scale the system. There are plenty of integrators and agencies that know Drupal well enough to produce a working site for you, but there are very few that know how to build one that will hold up with higher loads.

Content

Contributor Experience

Drupal 7 has greatly improved the usability of an interface that most contributors found to be overly complicated. At least visually, the difference between a visitor and an administrator is now more obvious: the administrator gets a complete toolbar on top of the page, whereas a visitor will only get “my account” and “logout” links. The system now also has a “dashboard” like WordPress. Unlike WP, however, Drupal's interface is an overlay on top of the website; thus, it would be impractical to open the UI with the dashboard. The dashboard is populated by blocks of functionality that a user can add and rearrange at will. Currently however, this is more of a promise; there are only a few widgets available, and they are no more useful than the regular menu items.

Several Drupal users we spoke to expressed their discontent with the “Overlay” module in version 7 that forces them to deal with a myriad of pop-up windows at every action point. Many overcome this annoyance by disabling the “Overlay” module completely.

Drupal 6's UI has been translated into more than 40 languages, but the translation efforts for Drupal 7 are still underway.

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

Contributing Content

In Drupal's out-of-the-box state, users with privileges to add content will see an Add Content menu item, which will take them to a page listing all the different content types installed on the system. Most content types have a body and/or description field that contains a text area for formatted text. Drupal has an input format filtering system that strips out potentially malicious text and does some text replacement (such as converting line breaks to paragraph tags). The core distribution ships with filters for Full HTML and Filtered HTML. Filters can be assigned to user groups based on trust. For example, anonymous users may have a filter that disables the anchor tag to prevent comment spam.

Most users will edit content with the assistance of a WYSIWYG editor. Surprisingly, Drupal doesn't come with a rich text editor — supposedly because it is a “pluggable platform” that doesn't mandate that you use a particular one. However, TinyMCE and FCKeditor are easily installed as modules. If you want more options, the WYSIWYG module allows you to pick any number of editors and easily configure them, turning specific buttons and functions on or off for full HTML or filtered HTML.

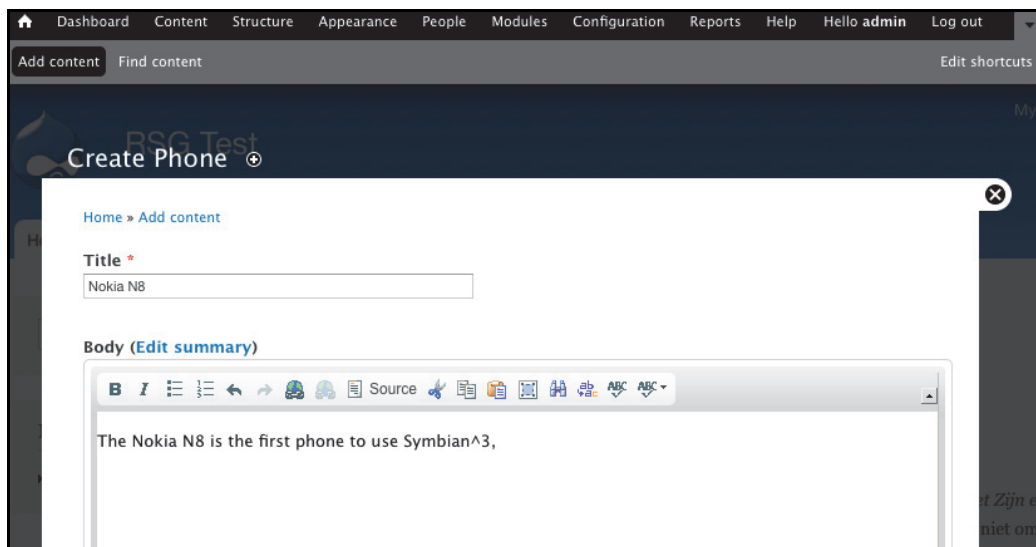


Figure 179. This is the FCKeditor in a sparse configuration in Drupal 7. Note the website in the background; the interface is an overlay on the site.

The flexibility comes at a price though: there is no tight integration of editor functionality. For instance, if you want to link to content in the Drupal site, you'll have to type in the node path yourself, because there's no link picker. In Drupal 6, this could be solved by a plethora of add-on modules further modifying the editor behavior, but so far, these are lacking in version 7. In WordPress, you may be stuck with TinyMCE (whether you like it or not), but at least it comes with niceties such as inserting images from thumbnails into the text by default. Drupal certainly looks a lot better in version 7, but it's hard to get the editing experience up to the basic level of other systems; you'll have to wait until more modules are ported from Drupal 6.

Unlike typical CMS tools that are based on a hierarchical folder structure, content is not “placed” in a site map, but is rather free floating, and is capable of displaying in many dynamic topical lists, based on a taxonomy system of vocabularies and terms. A vocabulary is a collection of terms that can be hierarchical or flat, controlled or free.

Drupal's approach to content organization is particularly in vogue right now, in light of the popularity of tagging and faceted navigation. While many hierarchically organized content management systems struggle to allow multiple locations of the site to share the same piece of content (you need to make a “link” or a copy), in Drupal this happens quite naturally by simply tagging the content with several relevant terms. For example, by tagging an article with the terms “WCM” and “Collaboration,” the article would appear both in lists of WCM articles and Collaboration articles. Note that this may represent an unacceptable mind shift for some authors used to “placing” pages on a website.

For more traditional site organization, a menu system allows the creation of a hierarchical tree of links (internal and external). Most content display themes use a default menu called “Navigation” for site navigation (in a left navigation, or with tabs across the top). Additional menus can be added and referenced by display templates.

Content Lifecycle

While WordPress has a simplistic notion of content lifecycle, Drupal has none at all. All node types possess an attribute called “Published,” which can be true or false, and that's it. As with everything, it is possible to build this out using modules, but so far, they have not been ported to Drupal 7. When they are, it won't be possible to save a new revision of a published piece of content without making the updates live. In sum, Drupal is not ideal for enterprises with formal approval, retention, and auditing needs.

Experience

Publishing

Out of the box, Drupal's URLs are query-string based and non-search engine friendly. However, Drupal comes with an aliasing capability where a page can be assigned a readable URL, but there is no central place to manage these aliases. By enabling mod_rewrite in Apache, and enabling “clean URLs” in the administrative interface, Drupal can reference pages without the query string. The Path module gives all nodes a URL alias field where a user can enter the path for the asset. The module Pathauto automatically generates aliases for nodes, categories, and users based on the name. These mappings are stored in the database and are updated in batch mode, rather than every time an item is published. This means that an updated file will not get a readable URL until a scheduled batch is triggered by a cron job on the server, which is far from ideal.

Digital Marketing

Although Drupal offers no formal personalization services, you can assign permissions regarding what content appears for which roles, to implement basic segmentation.

Experience	
Publishing	
Standards Adherence	
Multichannel	
Mobile	
Digital Marketing	
Site & Campaign Analytics	
Testing & Optimization	
Segmentation & Personalization	
Social Media Integration	
Promotional Campaigns	
Community & UCG	
Workplace	
Collaboration & Networking	
Dashboard	
Ancillary	
Site Search	
Online Forms	
Module Ecosystem	

Drupal specializes in collaborative content, and unlike most systems in this report (which offer only relatively basic features), Drupal's are quite strong. For example, there is a comment moderation interface, where a user can moderate all of the comments that have been added to the system. There are views that show recent content entries that can be used by a site moderator to remove offensive content. Users that are bending the rules can be tracked and blocked from adding content or put on probation, so that everything they submit goes into a moderation queue.

There is also a spam-filtering service, Mollom, which is essential for sites that allow commenting on external, visitor-facing content. Mollom was also founded by Dries Buytaert, and can be implemented in Drupal using a plugin. The service is also available for other systems, such as Joomla!.

Ancillary Services

The integrated full-text search functionality is adequate for searching text-based content, though file-based content is not indexed — making Drupal arguably less useful on an intranet. The default search configuration has a basic search and an advanced search that can look for keywords, exact match phrases, and can restrict by content type. Searches tend to return too many results rather than too few. Note that the index is refreshed by a scheduled script running on the server, rather than every time the content is updated. On the whole, this is quite weak.

Acquia Drupal offers Apache Solr as an alternative, but this is a SaaS solution, so you may not want to employ it on private sites (such as intranets or extranets). On the upside, Acquia's module leverages content tags, so you can display a faceted search interface based on these.

Unless configured otherwise, Drupal invites visitors to engage with the content actively. Drupal comes with multi-user blogs, basic polls, and forums. These have been improved by other add-on modules, although (as elsewhere) all but the most popular ones are awaiting a port to Drupal 7.

Project Intangibles

The Drupal community is large, active, passionate, and global — but loosely organized. The groups.drupal.org website is a good resource to find a local Drupal group, as well as topical groups that meet virtually through threaded discussions. There are periodic, well-attended Drupal conferences held throughout the year, most notably the twice yearly DrupalCon (one in North America and one in Europe). Drupal contingents frequently can also be found at various open source conferences.

A “Drupal Association” was formed at the end of 2006. Unlike similar associations in competing projects, the Drupal Association focuses on supporting and promoting Drupal, rather than making decisions about the direction of the project. Also, the Drupal Association does not own rights to name or the drupal.org and drupalcon.org domains. Those belong to Dries Buytaert, who grants permission to the Drupal Association to use them. The code is owned by the original authors who must license their works under the GPL (v2). All modules must likewise be GPL licensed. Unlike Joomla!, Drupal never had an exception for

Intangibles	
Vendor Services	
Vendor Professional Services	●
Channel Partner Services	●
Support & Community	●
Strategy & Roadmap	●
Viability & Stability	●

commercial modules and is not at risk of having the rift that was created when some Joomla! core team members spontaneously revoked the exception.

Today, governance is fairly tightly held. Technical decisions are made by a small team of core committers. While there are a number of long-time developers in the trusted circle (there are 14 maintainers with different responsibilities), the Drupal project revolves around Dries Buytaert similar to how the TYPO3 project used to revolve around Kasper Skårhøj. However, while TYPO3 tries to put more organizational structure in place to change that, Drupal seems to enjoy informality and quirkiness. For example, the release schedule has historically been inconsistent and unpredictable. Questions about release dates usually get rebuffed with responses such as, “When it’s done.” However, security patches are generally prompt.

Acquia the company isn’t as tightly tied to Drupal as Hippo (the company) is to Hippo (the CMS); Acquia Drupal is not commercial open source. Instead, Acquia’s revenue model revolves around specialist consulting and SaaS flavors of Drupal. However, the company does sponsor some development. For instance, Acquia hired usability experts to improve the user friendliness of its version 7 interface, and donated those improvements back to the community.

The difficulty of upgrading from version to version is variable. There is a tool on the website that enables you to select your source and target versions and returns a list of steps you need to follow. Most of the complexity happens at the database level, because the code is so modular. This is compounded by the fact that the core developers tend to favor improvements (even in the esthetics of the code) over maintaining backwards compatibility.

Modules remain the single biggest headache when upgrading Drupal. Modules are versioned according to the version of the core with which they are compatible. It is up to the module developer to specify what versions of the core his module supports, and the module developers are not always right. Drupal module developers also have perhaps too much freedom in creating database tables and writing to the file system.

There is no standard framework for these services, so there tends to be a good deal of inconsistency, which becomes difficult to manage. For example, if you want to change where binary files get written to the file system, there is no one place to do it, because each module has its own setting. Another example is that module developers are on their own to create variations of their SQL access code to work with different database back-ends. Partly because of this, many Drupal implementations are still running older versions – even with the advent of version 7, version 5 and even 4 implementations remain. In order to upgrade to version 7, current users must switch to version 6 first. However, version 7 makes large changes to (among others) the database and templating model; don’t expect an easy upgrade, and for the immediate future, don’t expect to have all of your modules for the new version.

There are a few small consultancies that focus on Drupal integration and support, but the vast majority of Drupal developers are independent contractors. The dearth of major, established consultancies here could present limitations for larger enterprise customers. Indeed, for larger implementations, Drupal’s current success means it has been hard to get specialist help in recent months. Nevertheless, Drupal is an application that a decent PHP developer can often self-support — unlike more elaborate and complex systems such as Plone.

Occasionally, individual consultancies or contractors offer Drupal training courses and there are plenty of professionally written books on Drupal. In fact, books on Drupal 7 started coming out well in advance of the actual release. The most useful resources are the online

handbooks, which have recently been reorganized, as well as the forums. The IRC channel is also a good place to ask a quick question and see the community interact.

If you want more reliable support, Acquia offers four levels of support subscriptions:

- “Basic” — \$349/year
- “Developer” — \$995/year
- “Professional” — \$2,500/year
- “Enterprise” — \$8,000/year

The enterprise level guarantees a 2-hour response time on tickets 24/7. Note however, that Acquia will only support version 6.x and up; on the other hand, for those versions, the company will also support any add-on modules you may be using. Another alternative (especially if you’re in the U.S.) is to utilize Acquia’s Amazon-based hosting and let the company manage your implementation and deal with scalability headaches. Pricing depends on configuration and traffic, but runs from the hundreds to several thousand a month. (According to the company, currently some of the largest hosted implementations range around \$70K – \$90K per year.)

Conclusion

Drupal's real power comes with community-oriented publishing and communication, and as a dynamic web application development framework. The availability and diversity of Drupal modules enables a site administrator to be opportunistic and experimental about adding new features the user community may embrace. Technology wise, Drupal's key strengths are its lightweight core with extensibility through the use of its modules. With the change to Drupal 7, however, you'll have to wait for modules to be readied for the system. It will take a considerable period for Drupal 7 modules to be remotely as plentiful as they were with Drupal 6.

Drupal's primary weakness is its lack of support for real configuration management, which becomes a challenge for large development teams or multi-environment implementations. It's no surprise that “I Hate Drupal” forums are dominated by two groups of critics: experienced developers frustrated by the limitations of the platform, and novice contributors overwhelmed by the complexity of its editorial interfaces. Drupal 7 offers some remedy for both, with a more uniform and abstract infrastructure and a more usable interface. However, these changes aren't as radical as the Drupal community would have you believe, and they don't change the essence of the system.

Drupal was simply not designed with traditional enterprises in mind, and when developers bump up against these shortcomings, it is not on anyone's priority list to fix them. Perhaps Acquia will take ownership of these issues and correct them, but the community remains very diverse and independent minded, so that road may not be easy. Acquia itself seems more interested in social software applications than it is in straightforward web publishing scenarios.

So, if Drupal's community emphasis fits well with your business objectives, the package is worth investigating, but don't assume that it will become “enterprise ready” in the near term.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

EPiServer: EPiServer CMS

episerver.com

Vendor at a Glance

Specsheet	EPiServer: EPiServer CMS 7 Summary
Geography	Europe, especially the Nordics, with a growing US presence
What's New	<ul style="list-style-type: none"> • Redesigned, cleaner user interface with drag-and-drop and in-page editing capabilities • MVT and A/B testing, campaign management, and other new digital experience features as part of the core CMS
Strengths	<ul style="list-style-type: none"> • Comparatively clean, Microsoft-based architecture, well suited to those customers with ample .NET developer resources • SharePoint integration will have strong appeal for customers who are already heavily invested in that platform • Company continues to exhibit strong technology vision and remains highly committed to latest Microsoft standards • Page/folder-oriented information model will feel familiar to many customers • Company focuses on core technology and leaves sales to resellers • Comparatively advanced digital marketing tools, including rules-based personalization, marketing dashboards, analytics, and ecommerce
Weaknesses	<ul style="list-style-type: none"> • Licensing model can get costly in a hurry; numerous modules and packaging bundles are very confusing • More of a development platform and less immediately “out of the box” than some other systems, requiring development in Visual Studio • Busy and inconsistent editorial interface requires contributors to navigate many tabs and pop-up menus • Power-user orientation might be overkill for some customers • Page-centric approach may not work well for licensees looking for more component-based content reuse • Only available via resellers, and they are comparably scarce beyond the UK and the Scandinavian home markets; growth in North America has been slow and uneven
Potential Fit	Microsites & Landing Pages, Basic Digital Marketing, Community Oriented sites
Unlikely Fit	Informational, Digital Workplace
Compare To	Ektron, Telerik, Kentico, Sitecore, SDL, Microsoft
Operating Systems	Microsoft Windows Server
Repository	Databases: Microsoft SQL Server, Oracle
Client	Browser
App Platform	.NET

Specsheet	EPiServer: EPiServer CMS 7 Summary
Licensing	Server and site based; Enterprise license starts at \$17,900, with staging and testing tiers priced separately
Ownership	Privately held; 220 employees, estimated \$42M in 2012 revenues

Summary

EPiServer is a mature, technologically up-to-date offering from a well-established Swedish company with aspirations that reach far beyond its home market. While the vast majority of EPiServer customers are in the Nordic region or elsewhere in Europe, the company is making strides to expand to new geographies, having established offices in 11 countries.

EPiServer CMS is a somewhat elaborate, .NET-based product that provides both content production and content delivery. The product’s history is clearly targeted at developers as a kind of overgrown SDK (Software Development Kit) for content management application development, more than it is an out-of-the-box solution. Within this context, however, it does what it does quite well, and appropriately resourced customers can expect to build powerful systems with it. In the past few years, EPiServer has added more marketing functionality, but the fact remains that you cannot take full advantage of the system if you don’t have sufficient internal or external IT resources.

EPiServer could provide a good fit if your enterprise has already made a significant investment in Microsoft technology and can leverage in-house .NET expertise. The company sells a variety of micro-applications, such as Community and Commerce, but these modules have diverse levels of maturity. You are more likely to be able to create and manage a small-to-mid size community site on EPi than have enough feature support for large and complex community-driven websites.

Note that while the vendor has international ambitions, the availability of third-party and peer support is primarily centered in Sweden and the Nordics. The company has only recently developed a noteworthy North American partner channel, rounding up more than 630 partners worldwide.

Introduction

Founded in 1994, EPiServer has been around longer than most of its competitors. In August 2007, founder Mikael Runheim stepped down as CEO of the company following the sale of a majority stake in the company to three venture firms. He was succeeded as CEO by Peter Larsson. Larsson became executive chair of EPiServer’s board in early 2010, when Martin

Scenario Fits	
Simpler Site	
Informational	<input type="radio"/>
Microsites & Landing	<input checked="" type="radio"/>
Mid-Range	
Basic Digital Marketing	<input checked="" type="radio"/>
Mobile Site	<input checked="" type="radio"/>
Community Oriented	<input checked="" type="radio"/>
Complex Site	
Advanced Marketing Portal	<input checked="" type="radio"/>
Global Enterprise	<input checked="" type="radio"/>
Multichannel Publishing	<input checked="" type="radio"/>
Ultra-Large Single	<input type="radio"/>
Digital Workplace	
SMB/Departmental	<input checked="" type="radio"/>
Global/Enterprise	<input type="radio"/>

Henricson was appointed as CEO to prepare the company for a public offering on the Stockholm Stock Exchange. The IPO, however, was withdrawn a few days before the trading began in late 2010. Soon after, EPiServer was acquired by IK Investment partners, a company known for buying and “prettying up” various businesses for further trading, albeit with scant experience in the technology sector.

Headquartered in Stockholm, the company has offices in Denmark, Norway, Finland, the UK, the Netherlands, South Africa, the US, Australia, UAE, Spain, and Vietnam. The company launched a Chicago office in 2009 (and now has 30 people in three US locations). Additionally, there are now 50 employees in Vietnam, dedicated to support, testing, and development.

EPiServer currently emphasizes competencies in several markets related to its core WCXM offering: community software, ecommerce, and digital marketing.

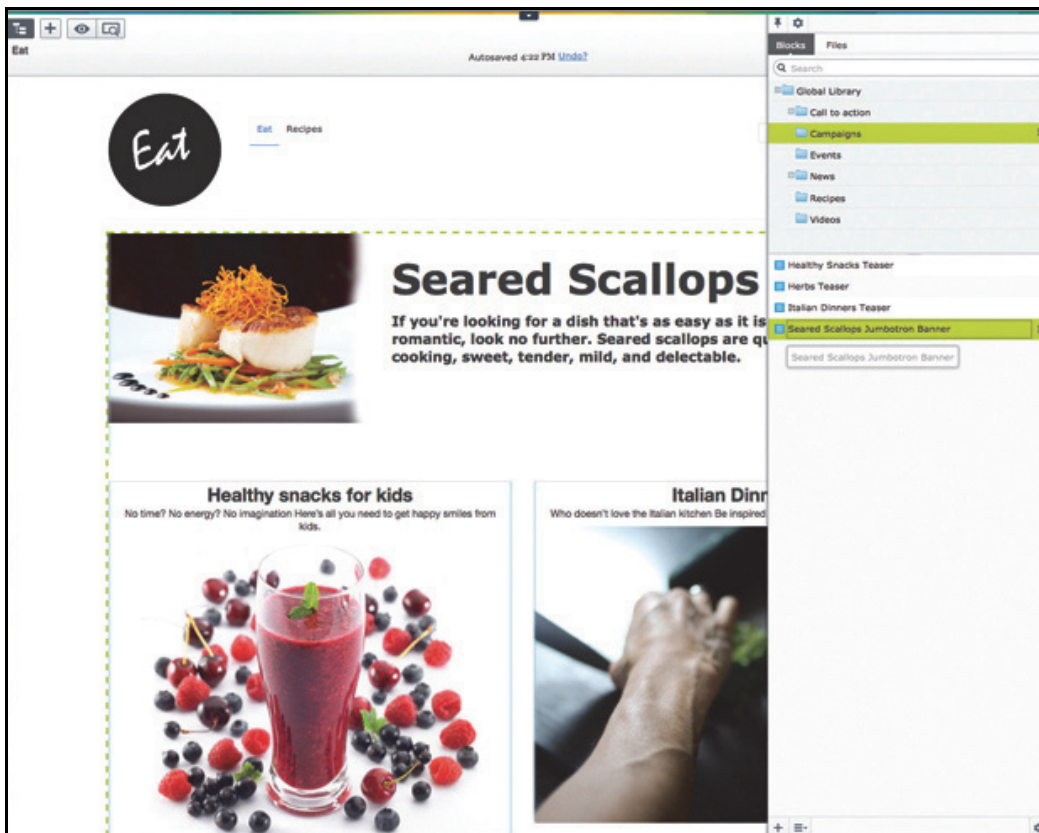


Figure 180. Authoring and managing content in EPiServer CMS using drag-and-drop functionality.

The origins of EPiServer’s community solution lie in the company’s April 2008 acquisition of Netstar. StarCommunity was re-branded as “EPiServer Community,” and StarMail became “EPiServer Mail.” However, while EPiServer Community is built on similar (.NET) technology, it is still a distinctly separate offering with only relatively superficial hooks into EPiServer CMS. The company separately offers “Relate,” which includes the CMS, the community software, and the templates to piece the two together.

Along the way, the nomenclature in EPiServer’s portfolio has expanded to epic proportions. There’s a plethora of EPI-centric names floating around, and it’s hard to tell whether they refer to some specific product, particular parts of it, or some kind of partnership. For existing customers, this may represent a gradual shift, but for others, it becomes hard to distinguish among Relate, EPiServer Commerce, or EPiServer World, to name a few.

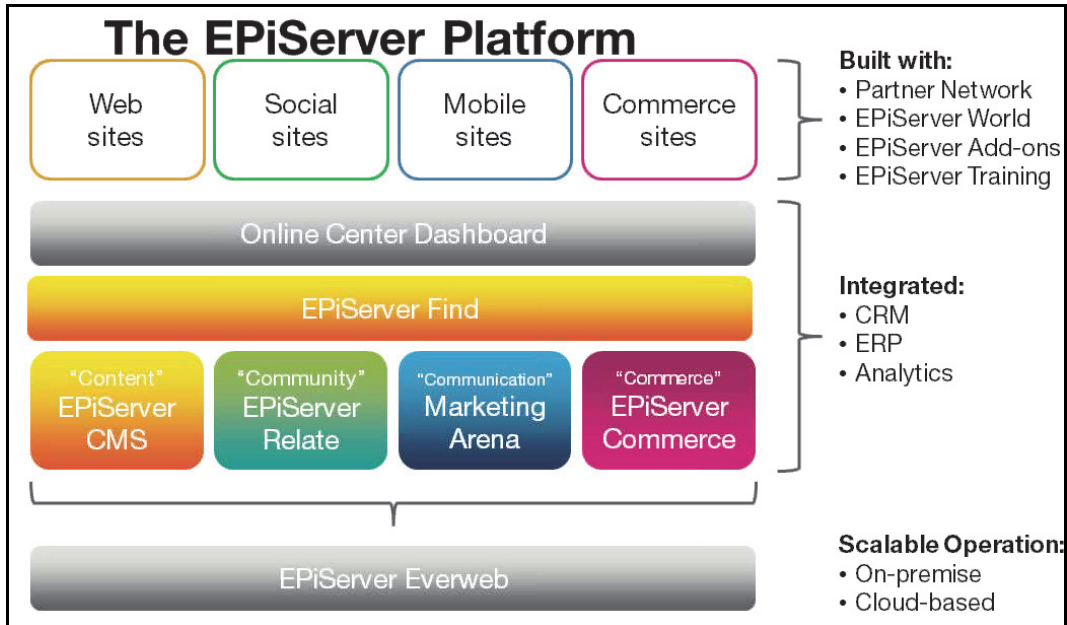


Figure 181. The EPI universe as presented by the vendor. Source: EPiServer.

The company itself seems to underestimate the confusion this creates, so let’s try to categorize some broad distinctions:

- **Products** – Various “product suites” consisting of the core CMS, and specific additional variations, such as Commerce and Relate
- **Modules** – Add-ons that are designed to address the needs of specific vertical markets, such as EPiServer Mail, and Google Analytics Add-on and Social Reach Add-on
- **Connectors** – Allow EPiServer CMS to interact with other systems, such as EPiServer Connect for CRM, and EPiServer Connect for SharePoint
- **Communities, partnerships, and solution marketplaces** – Such as EPiServer World and EPiCode

Let’s have a look at each in a little bit more detail.

Products

Currently, there are three flavors, each consisting of the core system and additional modules and templates. These products are designed to address what EPiServer has identified as important scenarios:

- **Relate** – For communities and social collaboration, including a set of templates, tying in EPiServer Community 4 and EPiServer Mail. There is also a Relate Intranet Edition, which adds EPiServer Connect for SharePoint

- **Commerce** – For product catalog management and customer management, which is based on technology from a company called “Mediachase,” which was acquired by EPiServer in 2012 (following a prior partnership)

Modules

EPiServer offers various optional modules (though these may be included in specific packages):

- **EPiServer Mail** – Formerly StarMail, for email campaigns
- **EPiServer Find** – Big data product designed for content retrieval applications and onsite search

Connectors

The company currently has two connectors, launched in November 2008:

- **EPiServer Connect for SharePoint** – An already existing connector, now in version 2, it allows relatively tight two-way integration of EPiServer CMS and SharePoint
- **EPiServer Connect for CRM** – For connecting to Salesforce or Microsoft CRM

EPiServer CMS’ development platform-like nature means most rollouts entail customization. EPiServer’s partner channel relishes this, of course. Like its competitor Sitecore, EPiServer’s strategy of outsourcing all sales to its partner channel has resulted in a fairly successful business symbiosis.

EPiServer also offers a hosted “Platform-as-a-Service” version of its content management system, which is delivered on a virtual server that EPiServer manages through its Everweb hosting service, running on Rackspace in Europe.

Again, EPiServer doesn’t make licensing easy; a contract for EPiServer Cloud is limited to up to 50,000 page views per week, 300 GB of data per month and 100 GB of storage. A starting monthly fee is approximately \$1,800.

Technology

Technical Administration and Security

The product has an unmistakable Microsoft bias, which is definitely a good thing for customers who have made significant investments in Microsoft products and product training. EPiServer’s somewhat elaborate technology stack leverages the .NET framework extensively (Figure 182). The product comes with its own repository.

Many customers run EPiServer on a single server — even those who use it for both web and intranet scenarios — although larger licensees are increasingly inclined to introduce staging environments for editors. (Unfortunately,

Technology	
Technical Administration and Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

EPiServer now charges extra for staging licenses.) Content can be published manually or automatically.

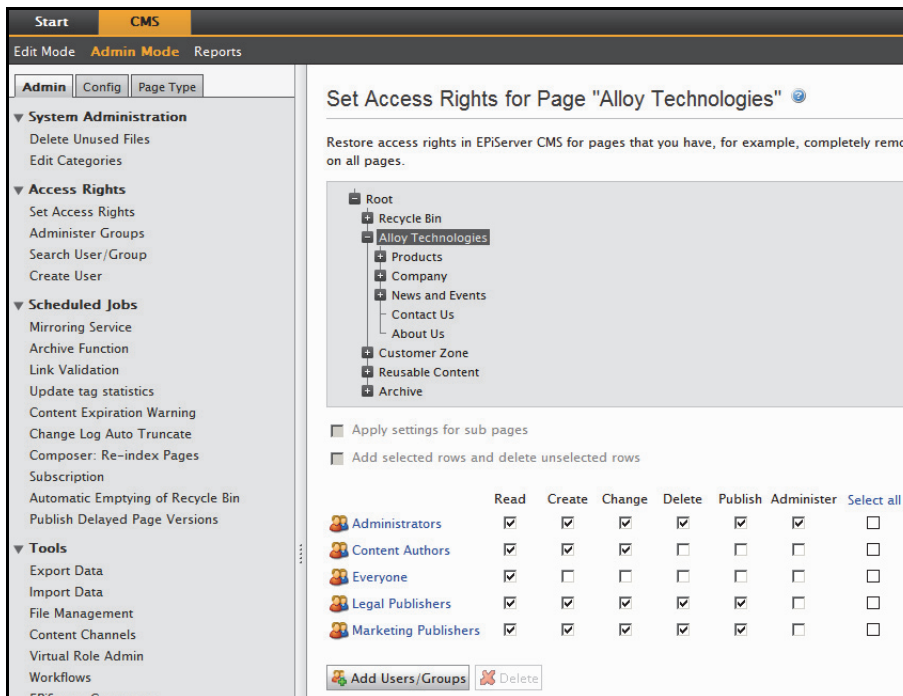


Figure 182. With flexibility in EPiServer’s roles-based security model comes complexity. Admins have to create and manage a multitude of variables: users, groups, roles, permissions to pages, types of pages, content items, properties, files, folders, and language variations.

EPiServer is rarely used by large customers with elaborate staging environment requirements, and this is reflected in the CMS and its architectural design. Unlike its more sophisticated competition, EPiServer doesn’t allow for streamlined and native publishing from the content management environment to a different server. Instead, you are bound to publish to a local server. To push content from staging to live, you must “mirror” that content, which is a more complicated setup since you need to pre-set “mirroring jobs” (using Windows Communications Foundation), rather than arbitrarily pushing sets of content items or sub-sites as you do with other content management systems.

EPiServer is a “frying” system. It delivers content using a dynamic, database-driven approach for certain bits of dynamically assembled page content, but a static approach for the rest. Note that it is not an n-tier architecture. You must run EPiServer on each front-end webserver.

User management is supported using Active Directory or any LDAP-compliant, third-party directory. As of version 5, it no longer stores users in the EPiServer repository, but lets you designate a separate store (RDBMS or LDAP) for users. The product conveniently offers multiplexing here via an extension to the .NET membership provider framework.

On the downside, the system supports groups, but — astonishingly — not roles. This will lead to a clumsy proliferation of groups and for larger customers, that could become very difficult to manage. Elsewhere, the permissions feel backwards; you apply rights on a folder basis,

rather than at the group level. You also manage group and folder properties in different interfaces. This is not an ideal system for a largely distributed team.

Like many other systems, EPiServer has disappointingly few reporting features. There is a logging system, but this is mostly technically oriented and mostly tracks errors.

Development

Developers create templates in Microsoft Visual Studio. However, EPiServer offers browser-based configuration modules where you can perform certain functions on templates, e.g., adding variables, inserting fields, and ranking. You should expect that most template changes will involve working with Visual Studio; if you do not have internal technical resources, this essentially means that your development partner needs to get involved every time you wish to change a template significantly. In either case, this is not a designer-friendly tool.

Overall, EPiServer takes a page-based approach to content, which means that you need to evaluate this approach carefully if you plan on heavily reusing content across multiple sites or delivery channels.

It is possible to create independent content objects using the product’s notions of “Object Stores.” Note however, that few customers have done this, perhaps because it is not particularly easy to do. The “Dynamic Data Store” in version 6 might remedy this for simpler tasks, but we’re not very optimistic about this at this time.

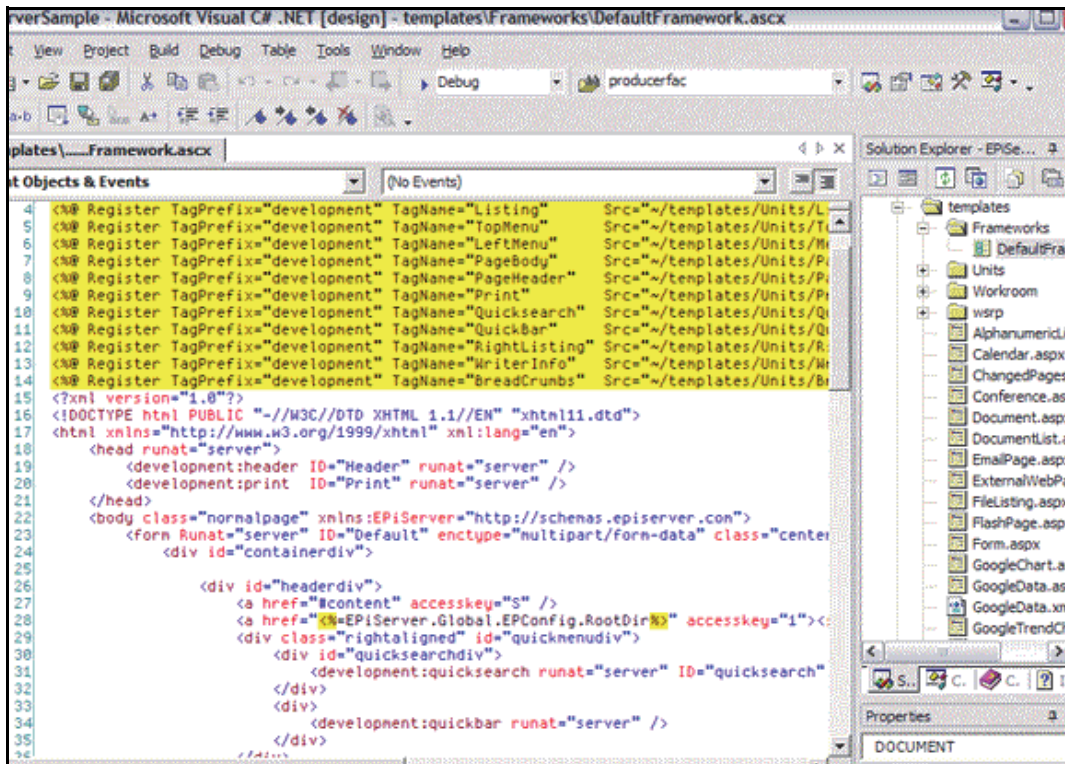


Figure 183. Developers create templates in Microsoft Visual Studio.

The base classes are implemented according to a “composition” design pattern. Developers can choose which functions and properties they want to use from the base class, instead of inheriting everything from that class when creating new page types.

Performance

EPiServer has a good reputation for scalability and performance, but you’ll want to examine the product’s capabilities yourself here. Customers who expect to have sites with high loads and high performance requirements should plan to simulate a workload and then do performance profiling to identify potential bottlenecks early in the project. EPiServer supports load balancing or mirroring, as well as distributed caching on multiple servers, even across geographies.

With respect to “cloudability,” some customers have deployed EPiServer in Azure (expensive) and Amazon EC2 (potentially flaky), but neither are officially supported.

The switch to .NET 3.5 and .NET 4.0 has facilitated improvements in caching, but again, no two customer scenarios are ever alike; plan on some careful testing here.

Content

Contributor Experience

EPiServer comes with a browser-based editorial environment that is oriented around tabular navigation, which enables you to access different parts of the system and any optional modules. EPiServer’s in-page editing feature allows authors to edit directly in-context, with the underlying TinyMCE rich text editor (Figure 184). Adding new content in this interface is clumsy though, and contributors will really need to understand (and navigate) the back-end tree structure.

Customers generally report that they are happy with EPiServer’s usability, from a contributor’s standpoint, despite small annoyances — like too many tabs and pop-up dialog windows.

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

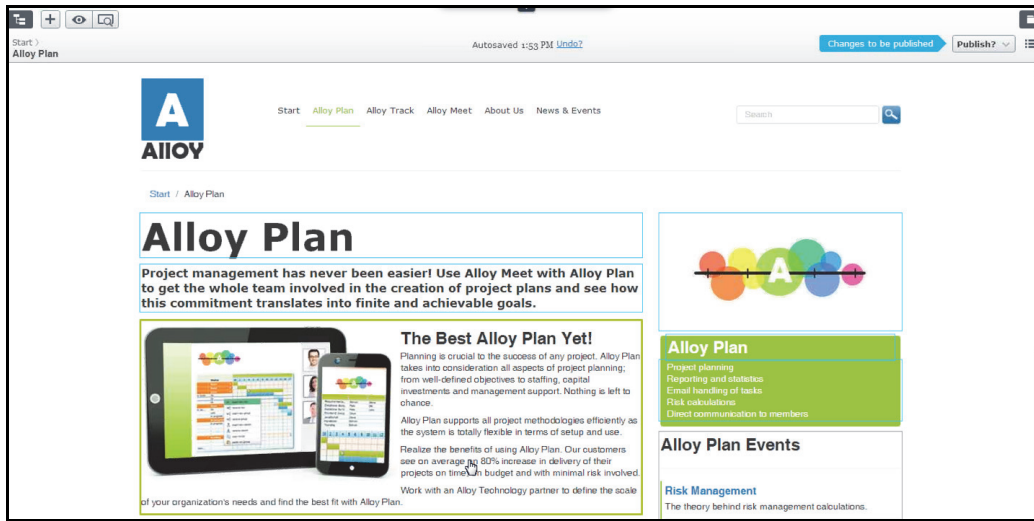


Figure 184. In-page editing experience in EPiServer, where editable content chunks are highlighted.

However, there are some important usability details worth noting:

- The default authoring interface tries to make efficient use of space by dividing the UI into tabs, rather than providing very long scrolling forms, for example. Other vendors (such as competitor Ektron, or even OpenText/Vignette WEM) take this approach as well. Some customers are put off by the complexities of EPiServer’s interface, which at least one refers to as “Vignette in a prettier outfit.”
- Tabbed interfaces are controversial, however, in the usability community, and some contributors can get confused about how to use them in a browser context. As always, you’ll want to test with your contributors.
- EPiServer uses an inconsistent mix of pop-up windows, internal tabs, and internal frames. This may cause confusion, and internal frames (e.g., with functionality to the left and to the right of the main screen) may require a larger monitor, to avoid excessive and painful horizontal scrolling
- EPiServer’s editorial interface has been localized to several languages, and users can set their own preferences in the UI. The default install will contain 12 languages (including English, Japanese, and most of the major European languages). When it comes to the admin UI (a distinctly different area of the UI for administrative use only) note that it has only been localized in six languages: English, Swedish, Norwegian, Danish, Finnish, and Dutch. In theory, since the localization is based on an XML file, it would be relatively easy to add a new language yourself.

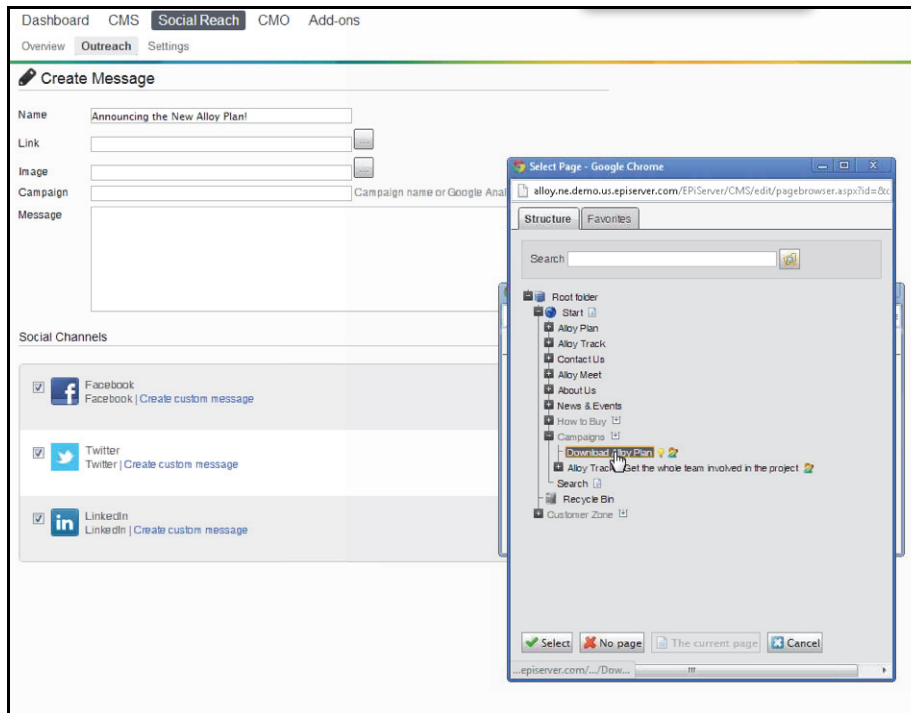


Figure 186. Configuring social messaging in EPiServer CMS. Note the mix of pop-up windows and internal frames.

Contributing Content

EPiServer allows editors to drag and drop content blocks within pages from the navigation tree. Starting in version 7, EPiServer made attempts to focus a little more on the notion of discrete content items, but this doesn't change the fact that the system's emphasis is primarily on pages, with some ability to manage content at a more granular level.

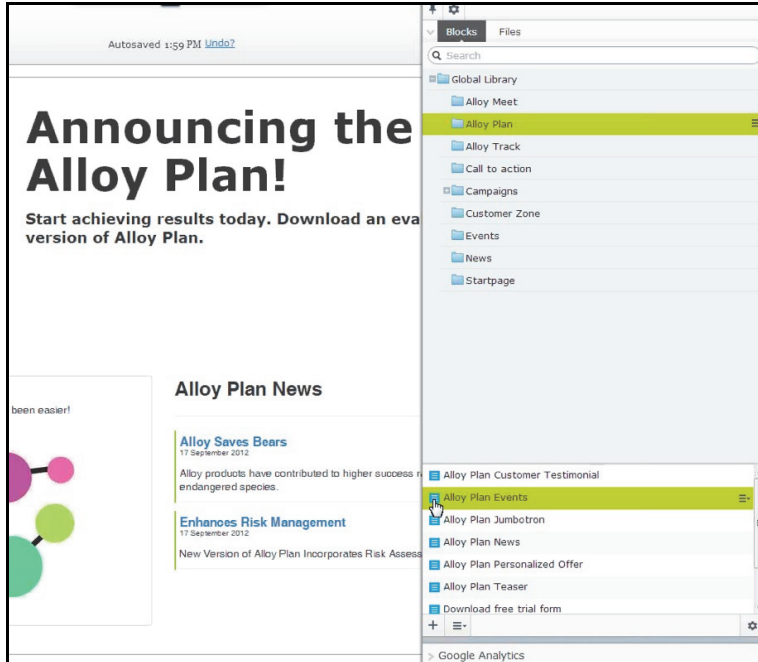


Figure 187. The editorial interface centers on a traditional “content tree” navigation. Despite the recently introduced notion of content blocks, EPiServer remains a page-based system under the covers.

The introduction of the “block” paradigm probably came from a desire to demonstrate more reusability with content chunks — somewhere that page-based systems (like EPiServer) traditionally do not excel. We recommend that you test this new paradigm very thoroughly if you rely heavily on content reuse across multiple sites, sub-sites, microsites, or pages. Take note of such limitations as the fact that only “global” blocks can be reused; reuse doesn’t apply to “local” blocks, which are always hard-wired to pages on which they’re placed. The only way to see what type of a block you’re dealing with is to check through the properties of every affected page.

For rich text editing EPiServer uses the popular TinyMCE. If you want to insert variable data in the WYSIWYG editor (e.g., user name, date, data pulled from a database via SQL query), you can use “Dynamic Content,” which works much the same way as field codes in Microsoft Word templates. Editors can embed specific tags into content elements and resolve them at run time to produce XHTML. Note that these tags are particular to EPiServer, and are not portable to other systems.

Preview in EPiServer is quite good. You can simulate different customer variables, including device types.

If you use EPiServer for lightweight document management, you can drag and drop files from inside Windows Explorer. EPiServer will store binary files (e.g., PDF files) that can be uploaded and linked from a page.

EPiServer has some light digital asset management features, since the browser UI allows for uploads of images, videos, and other rich media. It also offers basic image manipulation, e.g., scaling to predefined image sizes. The uploaded images are saved on the file system of the built-in Media Manager. This allows you to connect your assets library to SharePoint, Dropbox, or to ImageVault for further manipulation. For video assets, EPiServer supports Flash, Shockwave, QuickTime, Windows Media, YouTube, and RealMedia.

Finally, the system has a comparatively advanced and usable taxonomy management framework, and simple tagging fields (with auto-complete) for classification.

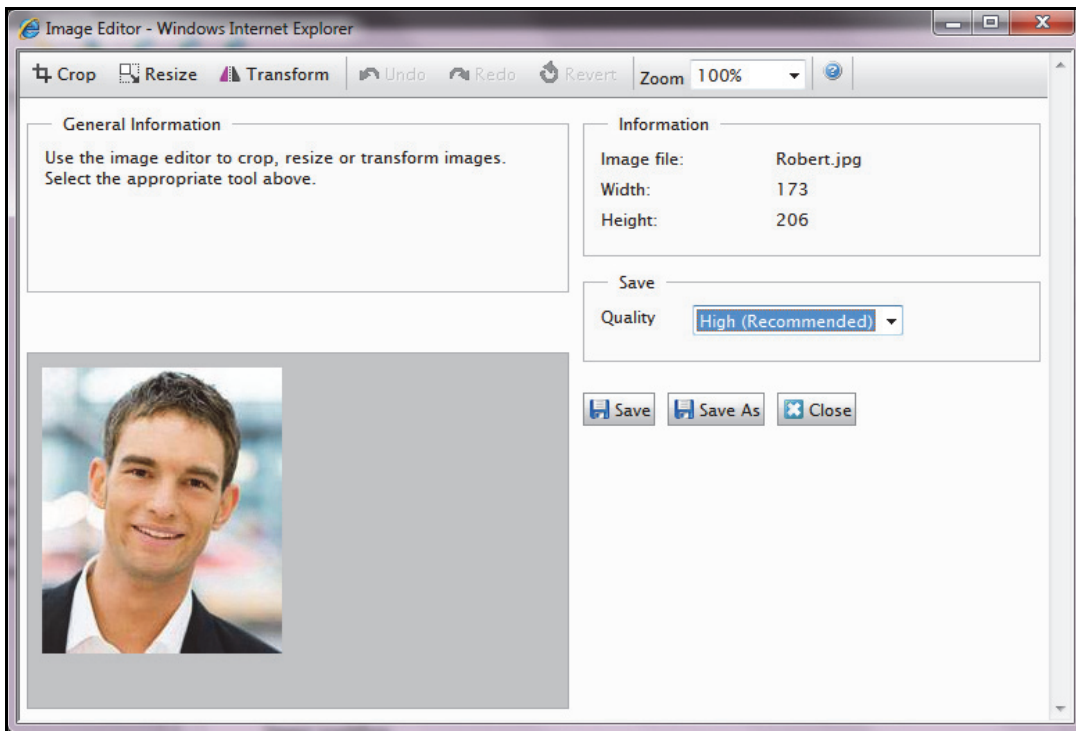


Figure 188. Image manipulation options in EPiServer include cropping, resizing, transformation — all done from within the interface of the EPiServer Image Editor.

One of the missing features in EPiServer is a meaningful Adobe InDesign integration, rendering this system a rather poor fit for web-to-print or print-to-web scenarios. The most you can usually do is partial ingests, but keep in mind that the resulting XML will most likely require laborious optimization for any useful outcome.

Content Lifecycle

At one point, EPiServer was building its own workflow engine, but the company stopped its development when Microsoft announced Windows Workflow Foundation (WWF). EPiServer started using WWF, instead, but that didn't rid the system of a rather clunky workflow implementation that is plagued by multiple pop-ups and a tab-heavy interface.

You can manage content in multiple languages, even down to property values per language. Unfortunately, EPiServer uses flag iconography to indicate language. While this works well in the Scandinavian home market, where each country has its own language, it's not always clear which flag to use when many countries map to a single language (or one country to several). Modules are available (through EPiServer Add-on Program) to integrate with Lionbridge and Translations.com.

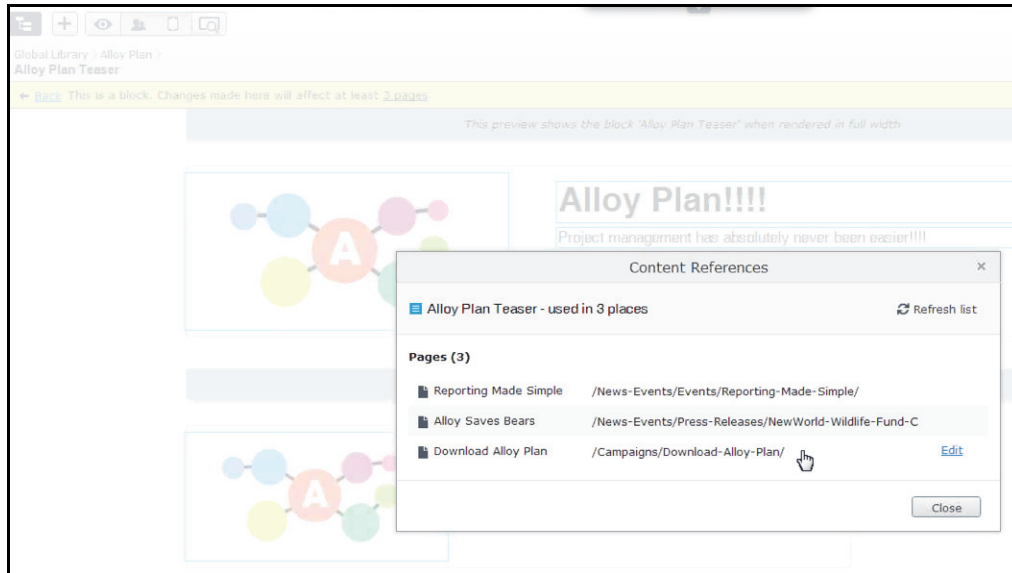


Figure 189. Recently, EPiServer started offering a “where used” functionality, which allows content editors to see all places/pages in the CMS that reuse the same content block. This is helpful if you need to know which pages will be affected when you change any content block.

Experience

Publishing

The product has a decent URL structure that follows the structure of the site. For example:

<http://www.episerver.com/en/Customers/>

The URL stems from the page title and once created, is stored in a separate field. However, non-English language customers should note that special characters (e.g., æøåüö) are simply stripped out by default and replaced with the closest ASCII equivalent (instead of escaping the characters with %-hex values). On the plus side, the URL will not change simply if the page title is updated. Note that the Friendly URL subsystem, or FURL (as it is called), supports bidirectional URL rewriting and has a well-documented C# API, should your developers want to customize the system. Links are managed and stored separately from other system artifacts in EPiServer’s repository for use by FURL, which is quite sophisticated.

EPiServer provides a good starting base for mobile content management and content delivery. You may need to customize or develop additional functionalities on top of what EPiServer

Experience	
Publishing	
Standards Adherence	●
Multichannel	●
Mobile	●
Digital Marketing	
Site & Campaign Analytics	●
Testing & Optimization	●
Segmentation & Personalization	●
Social Media Integration	●
Promotional Campaigns	●
Community & UGC	●
Workplace	
Collaboration & Networking	●
Dashboard	●
Ancillary	
Site Search	●
Online Forms	●
Module Ecosystem	●

offers, but out of the box, there is some support for responsive design and HTML5, device recognition, mobile targeting, and personalization.

Digital Marketing

Along with Sitecore and Ektron, EPiServer is an active player in the arena of web experience management. In recent releases, the company has focused on several enhancements to personalization, campaign management, social media, and visitor segmentation and targeting.

You can define “Visitor Groups” and break down your audience into meaningful segments, so that you can target each with more personalized content appropriate to their needs. As in other interfaces in the CMS, the drag-and-drop capabilities add a nice usability touch for marketers or non-technical users.

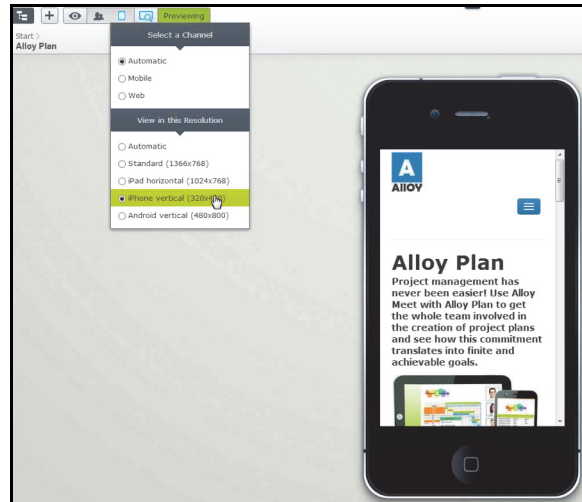


Figure 190. Mobile preview in EPiServer allows you to see what the content will look like in popular resolutions and dimensions (e.g., iOS and Android) directly from the CMS.

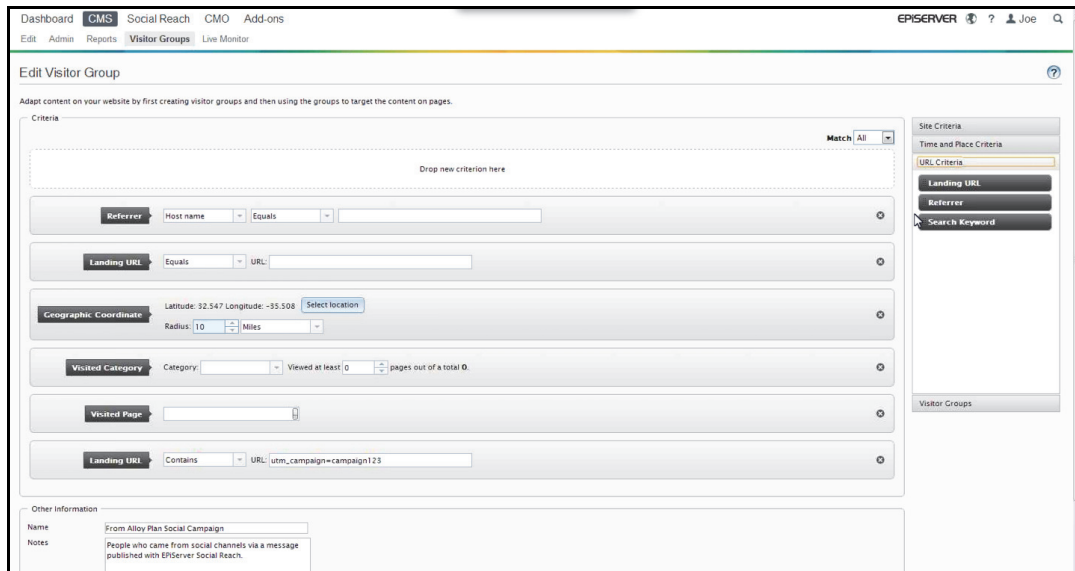


Figure 191. Visitor Group management in EPiServer CMS is a rather elaborate setup based on a multitude of variables — from referrer to “Visited Pages.”

Out of the box, you get several basic criteria for segmenting your audience: based on the IP address using their existing geo-IP database, number of visits, pages visited, referrers, and others. Beyond that, you will have to involve developers to create custom criteria.

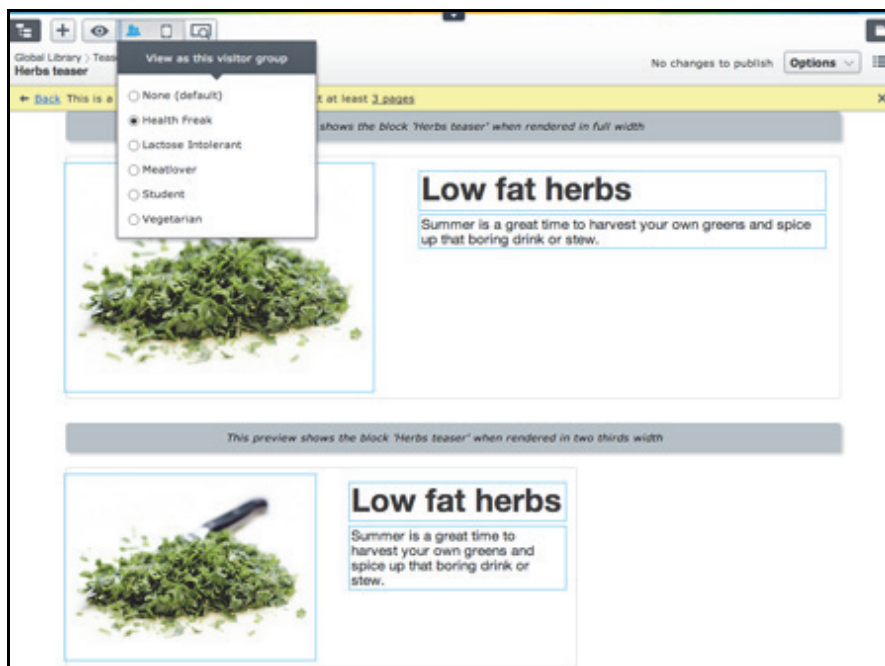


Figure 192. Personalized content can be delivered to various segments of visitors (called Visitor Groups in EPiServer) across various devices and channels.

Interestingly, you can use visitor traffic data to change the site (e.g., “five most popular products”). Using this approach, EPiServer tracks site usage and automatically creates listings of the most read articles or most popular products. These can then be updated as frequently as needed (e.g., daily, weekly, or monthly).

EPiServer also offers a feature called Campaign Monitor and Optimization (CMO) that contains two modules for digital marketing purposes. The Landing Page Optimizer allows you to do basic A/B and multi-variant testing (and determine the best alternative).

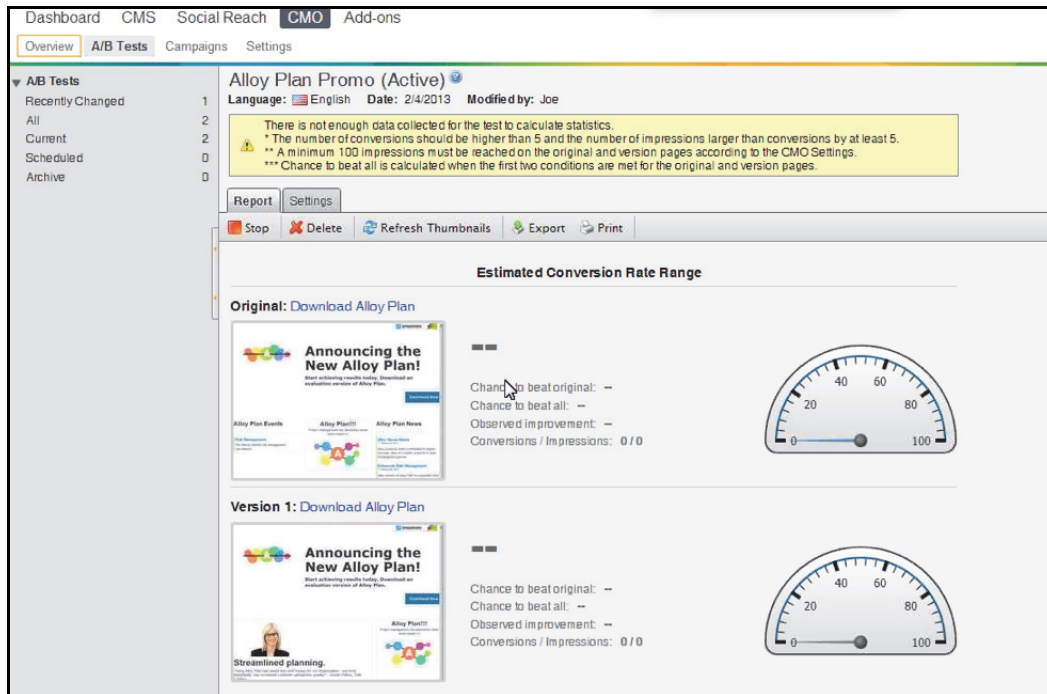


Figure 193. Setting up an A/B Test in EPiServer by assigning an Original Page and a variant of it to check which one will perform better based on certain parameters like the number of conversions.

The second module is Campaign Monitor, which allows you to test and monitor your marketing campaigns with some light reporting capabilities. Usefully, these two modules don't appear to have dependencies, and you can use either one — without paying for both.

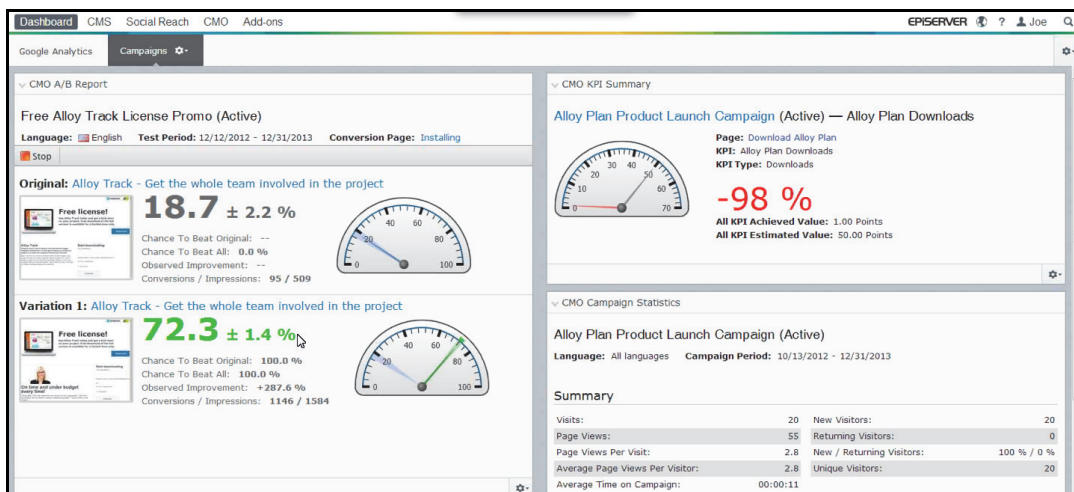


Figure 194. Tracking the performance of a campaign from EPiServer dashboard by using an A/B testing report, where Variation 1 of a page shows better results. In general, various dashboards in EPiServer CMO are probably one of the most attractive features for marketers looking to get a quick glance into campaign performance, but you don't expect much depth or detailed data breakdowns here.

The vendor readily admits that these tools are relatively simple compared to standalone products, and they are intended as an entry-level convenience.

Interestingly, though, EPiServer has adapted the KPI (Key Performance Indicator) term for scoring. Theoretically, this would allow you to have an “objective” measurement of improvements on your site. (Of course, if you don’t have in-house resources to go through continuous evolution of the pages, the score can be painfully static until you budget external work. For this reason, integrators can actually turn the scoring off for their implementations.)

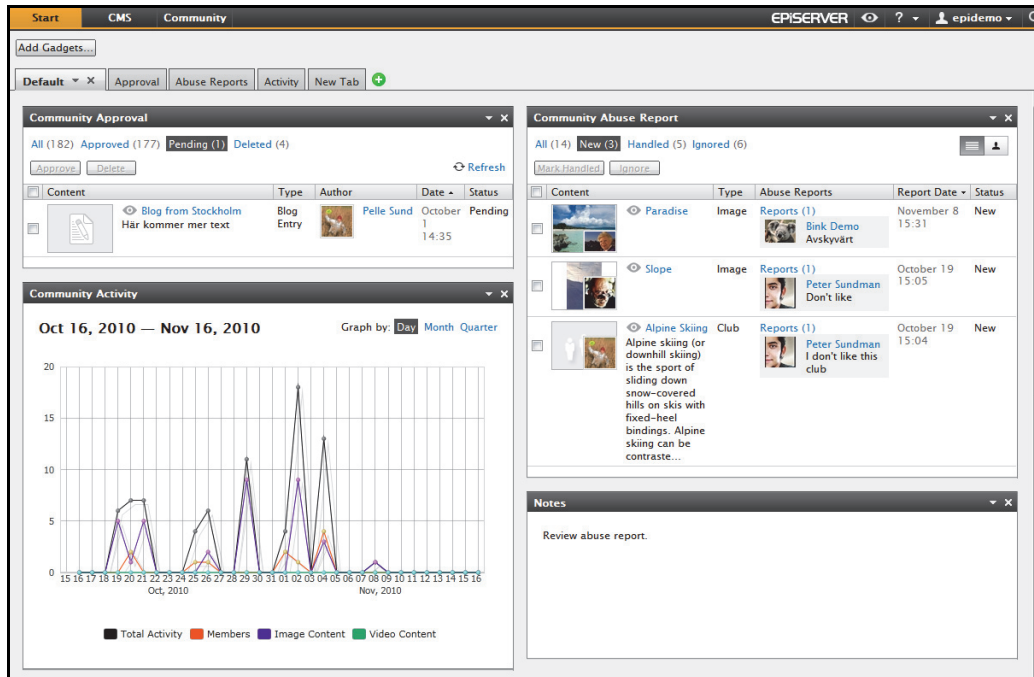


Figure 195. Blog comments can be moderated from within the EPiServer Community module.

Another module worth mentioning is “EPiServer Commerce,” based on [Mediachase's .NET e-Commerce Framework \(ECF\)](#) that was acquired by EPiServer in early 2012 after 1.5 years of OEM relationship. EPiServer Commerce can be integrated with the CMS after you download, install, and configure it as a separate package. Aside from the ability to manage your online store activities using this module, you can also do content, community, and customer management from a unified — albeit complex — UI.

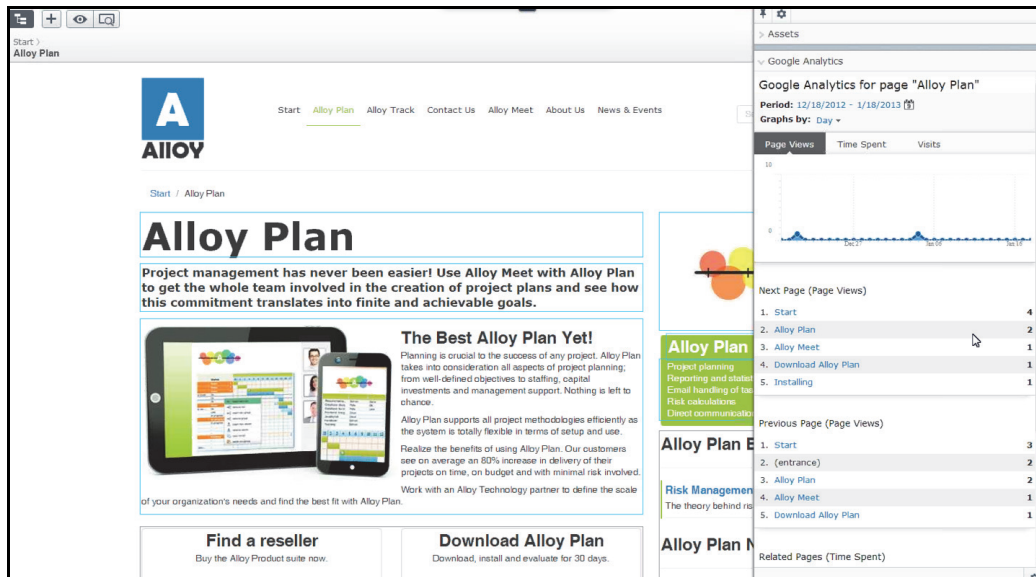


Figure 196. In-context and in-page access to Google Analytics is appreciated by content editors who need a quick glance into how their content is performing for further optimization if necessary.

Ancillary Services

Search functionality comes with the product and is nicely integrated with permissions so that users only get search results based on their privileges. You can also use the Google Search Appliance, integrated via a free module. The integration uses AJAX to populate the search results as you type, and again, you only see search results that you have permission to access.

In 2012, the vendor also launched EPiServer Find — an index-based search that can be used for site search or data-driven applications.

EPiServer offers an unusually strong, bi-directional WordPress connector for those who prefer that best-of-breed blog platform.

Like most of its competitors at this tier, EPiServer offers a fairly slick forms-creation and management subsystem.

Meanwhile, you can find a plethora of add-on modules from EPiServer itself, as well as third-party partners. Notice that the company will “approve” third-party add-ons, but you should not assume that they have been rigorously tested or certified, particularly with respect to security.

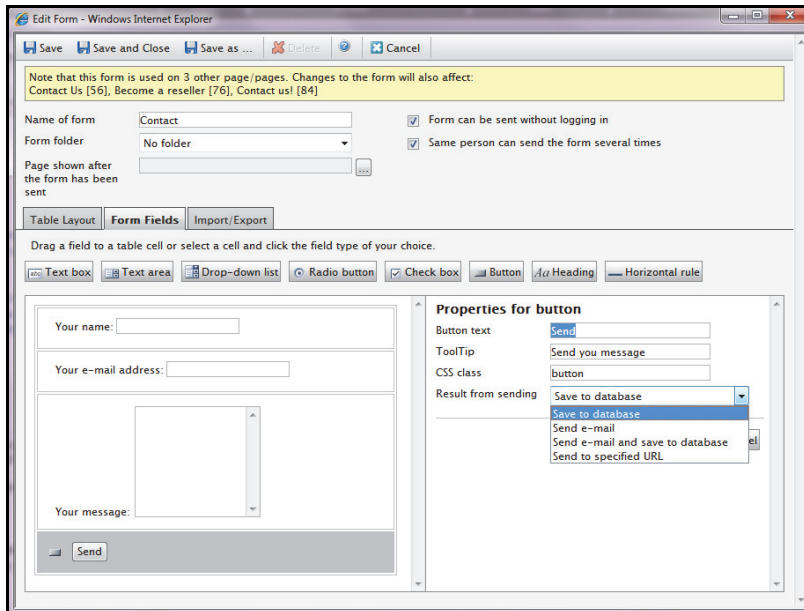


Figure 197. The form creation interface in EPiServer, with the W3C XForms standard behind the scenes.

Vendor Intangibles

The company has grown quickly in recent years and now counts over 240 employees, many of whom are based in Sweden.

However, it's worth remembering that rapid growth of the kind EPiServer is experiencing can be destabilizing for a company; where new levels of discipline in Finance, Operations, and "people management" are called for if the business is to remain profitable, and where retention of talent can become an issue if the attendant changes are perceived as bureaucratic. The ease with which EPiServer can meet these sorts of challenges remains to be seen.

EPiServer grew out of the founder's living room, and unlike many of its competitors, it prospered for years without any venture capital, right up until August 2007 when Amadeus Capital Partners and Northzone Ventures bought the majority of the company's stock. The original owner and CEO, Mikael Runheim, then stepped down. The new backers brought in Peter Larsson as CEO to grow the company. Larsson then became Executive Chairman of EPiServer's board in early 2010, when Martin Henricson was appointed as CEO to prepare the company for a public offering on the Stockholm Stock Exchange. Despite all of the confidence, growth, ambitions, and investments, the IPO didn't succeed, and it was withdrawn before it was offered. When the book-building period concluded (leading up to the first day of trading June 30, 2010), the offer was fully subscribed in the range of SEK 52-60 per share. But the actual first day of trading never happened; EPiServer decided to abort the IPO, citing "volatility" on the financial market.

Soon after, EPiServer was acquired by a surprising buyer: IK Investment Partners, a European private equity firm with very little experience in the technology sector. Financials for the deal were not disclosed, but we know that at the time of the IPO, EPiServer was valued at 600-700

Intangibles

Vendor Services

Vendor Professional Services	●
Channel Partner Services	●
Support & Community	●
Strategy & Roadmap	●
Viability & Stability	●

million SEK. It is not unreasonable to assume that IK Investment Partners has invested in EPiServer with possible future plans for selling it off.

The company reports 2012 revenues of \$42M. This data is hard to verify, since EPiServer is privately held.

A single-site EPiServer license retails at \$17,900 per server for the Professional license, which includes unlimited CMS users on a single IIS application. The EPiServer Enterprise license (multiple IIS applications + some additional features) starts at \$33,400, for “One Site Solution,” which runs on two servers. Additional servers are \$5,300 each and can be used for any purpose (load balancing, staging, etc.).

Customer and partner developer licenses are free. Additional flavors like EPiServer Social CMS (former Community solution) runs on two servers and is \$41,900, with unlimited users. EPiServer Commerce is \$43,500.

Maintenance and support for EPiServer CMS are charged at 20 percent of the license price and are handled out of the company’s headquarters in Stockholm (though the company says it can now offer global support through its US and Australia/New Zealand offices). Unlike most other vendors in the region, EPiServer also offers hosting. EPiServer in a hosted model starts at \$1,800/month for modest amounts of data transfer and storage; the price quickly skyrockets for more complex scenarios.

As previously mentioned, licenses are sold mainly through the company’s partner network, which also handles most implementations and first-level product support. However, there are instances when EPiServer leads a sales engagement. Partners typically receive as much as a 25 percent commission for the license sale, which is worth remembering in negotiations — particularly if the partner has also helped you choose the platform.

Make sure you can get competent resources on your project. An experienced partner network is available in the Nordics. EPiServer also runs a certification program that requires partners to train developers and measures status in three levels based on number of trained developers and license sales. Knowledgeable integrators beyond Scandinavia remain few and far between, and many “certified” North American partners still have very thin implementation experience. EPiServer itself has a small team of “Expert Services” to support partners in implementations.

As with other aspects, EPiServer attempts to juggle various strategies for creating an ecosystem to support the product simultaneously:

- **EPiServer World** – A community for developers and customers; it’s a website with documentation, instructional videos, and developer blogs
- **EPiCode** – An open source community, with a library of free templates and functionality to use in the development of an EPiServer CMS implementation. These come in the form of “PageBank” modules, which are sets of pages and code that enable, for instance, a MS Messenger Gateway or the WikiX wiki

Documentation is available in five languages, including English, Swedish, and Norwegian. These docs might be too technical for your non-power users (and maybe your power users as well), so consider producing your own “cheat sheets” for internal use. If you want to learn all the technical details on the new releases, note that training in the past has been open only to partners. EPiServer World, the company extranet, contains manuals, FAQs, and an online user forum with conversations in English, which seems to be quite active. There are now over 18,797 registered members for the site.

Conclusion

EPiServer has done little to further the product's visibility outside of its home market. With the firm focused primarily on technology and support, marketing has traditionally not been its strongest suit. However, this is slowly changing, and in the past few years the company has managed to find a modest following both in the UK and North America.

With CMO and the Commerce and Community modules, EPiServer is starting to direct its product more toward scenarios oriented for digital marketing and experience management — while also supporting and enhancing the core CMS product.

Buyers with strong, Microsoft-oriented IT departments should consider EPiServer as a solid alternative to offerings from other Nordic vendors, such as Sitecore. Architecturally, the system is comparable to Ektron and Kentico, but hews more closely to base Microsoft standards.

Prospective buyers outside of Scandinavia should temper expectations about support. You can find EPiServer integrators as far away as New Zealand, but solid expertise is generally tougher to find outside of EPiServer's home market.

Bear in mind that this is a fairly deep product technologically, and that it requires a certain amount of .NET development talent to set up, customize, and maintain. Developers have to work closer to the iron with this one. To be sure, EPiServer has plenty of tools it would like to sell you to add on to the core CMS — but as with Swedish furniture, some assembly is required.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Hippo B.V.: Hippo CMS

onehippo.com

Project at a Glance

Specsheet	Hippo B.V.: Hippo CMS 7.9
Geography	Primarily Europe, especially Benelux, with a growing footprint in North America
What's New	<ul style="list-style-type: none"> • More advanced personalization based on personas • Better architectural separation of content and delivery tiers • New, more standard ("CK") rich text editor
Strengths	<ul style="list-style-type: none"> • Hippo B.V.'s technical and support acumen is highly regarded • Java platform built off contemporary standards • Can potentially support Multichannel Publishing with decent mobile support • Standards-based architecture is highly extensible in the right hands • Company is making an effort to expand in North America • Well-documented Java and REST APIs
Weaknesses	<ul style="list-style-type: none"> • Java platform built off contemporary standards • Deliberately optimized for Multichannel and Global Publishing • Standards-based architecture is highly extensible in the right hands • Company is making an effort to expand its partner channel • Strategy of integrating with third-party systems is suitable for enterprises pursuing a best-of-breed approach to digital marketing communications • Hippo B.V.'s technical and support acumen is highly regarded • New personalization features show strong promise
Potential Fit	Multichannel Publishing, Global Enterprise
Unlikely Fit	Global/Enterprise Digital Workplace, Advanced Marketing Portal
Compare To	Adobe, Magnolia, OpenCms, Drupal
Operating Systems	Windows Server, a wide variety of Linux flavors
Repository	Apache Jackrabbit
Client	Browser: IE (8.0+), Firefox, Chrome, Safari
App Platform	Java: Tomcat, Oracle WebLogic Server
Licensing	Community Edition available under Apache License 2.0; Enterprise Edition is sold commercially at a median license cost of \$50,000
Ownership	Privately held — Hippo B.V. holds all significant copyrights; 90 employees

Summary

Hippo is a very engineering-intensive, Java-based platform that brings great flexibility to the developers that master its inner workings.

Throughout many years of product evolution, Hippo CMS has managed to garner adoption in several sectors — including government, publishing, financial, and education — particularly in the Benelux region.

Hippo offers a strong fit for scenarios that emphasize fine-grained content models for Multichannel Publishing (e.g., mobile), and is an increasingly plausible fit for content-heavy global web publishing operations. However, don't overlook the complexities of the platform, which has traditionally favored customization by an experienced developer over simpler configurability.

The commercial firm behind the project, Hippo B.V., is pushing the technology to become more marketing-oriented, but this remains new and rather uncharted territory for the platform. On the plus side, the platform will integrate with an impressive number of third-party services — a very useful attribute for savvier enterprises pursuing a more best-of-breed technology strategy for digital marketing and communications.

You may find it difficult to find developers experienced in Hippo CMS, particularly in North America, as the vendor's journey to "cross the pond" remains incomplete. A strong (if smallish) support community can help mitigate this problem. Hippo B.V.'s own commercial support is well regarded — something of a rarity in this space.

Over the years, Hippo has implemented many examples of cutting-edge Java and Apache technologies within its product. Make sure that you have skilled developer talent available in order to exploit them properly.

Scenario Fits	
Simpler Site	
Informational	<input type="radio"/>
Microsites & Landing	<input checked="" type="radio"/>
Mid-Range	
Basic Digital Marketing	<input checked="" type="radio"/>
Mobile Site	<input checked="" type="radio"/>
Community Oriented	<input type="radio"/>
Complex Site	
Advanced Marketing Portal	<input type="radio"/>
Global Enterprise	<input checked="" type="radio"/>
Multichannel Publishing	<input checked="" type="radio"/>
Ultra-Large Single	<input checked="" type="radio"/>
Digital Workplace	
SMB/Departmental	<input type="radio"/>
Global/Enterprise	<input type="radio"/>

Introduction

Hippo B.V. is an Amsterdam-based software development company founded in 1999. The company built what was originally a commercial CMS of the same name, but in 2004 released the code to open source as part of a shift in business model to emphasize product support and development over consulting and integration services.

The company co-founders had strong roots in the Apache community and remain active there today. They built the previous versions of Hippo the product largely around key Apache projects like Cocoon, Slide, and Lucene. At one time, the company also supported "Hippo Portal" as the official delivery tier for the platform. A commercial version of the original Apache Jetspeed Portal, this offering was discontinued in favor of a Hippo CMS delivery tier described in the next section.

The company earns income primarily via paid support subscriptions, and subscription-based license fees to the “Enterprise Edition,” which has additional functionality. In this evaluation, we review that the Enterprise Edition.

Version 7.0 was released at the end of 2008, and constituted a complete rewrite, installing more contemporary and transparent repository and appdev frameworks. Slide was replaced by Apache Jackrabbit, the Java Content Repository also used by Adobe, and Magnolia. The CMS interface is currently built in Apache Wicket, but slated to move to Angular JS. Lucene is used for search. Version 7.9 came out in July 2014, and with the earlier 7.8 release, it represents a bit of a turning point for the company in its emphasis on marketing-oriented features.

Today Hippo has about 90 employees, most of whom work from the Amsterdam headquarters and a Boston office.

Technology

Technical Administration and Security

The Hippo CMS architecture is based on the Apache Jackrabbit (JCR) repository as the pivotal core for the system; all content is stored in the repository, and all content management functionality is implemented there, as well.

Amongst supported application servers, you have a wide range of options — customers have implemented Apache Tomcat, IBM WebSphere, Oracle WebLogic, and JBoss — but only Tomcat and WebLogic are officially supported.

To work directly on the repository, Hippo provides the “console,” which is a JCR browser that allows you to look through the repository and add or edit values there. (Windows power users may be familiar with the registry editor, which is somewhat similar.) It’s useful to be able to perform these tasks from the relative comfort of your browser.

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

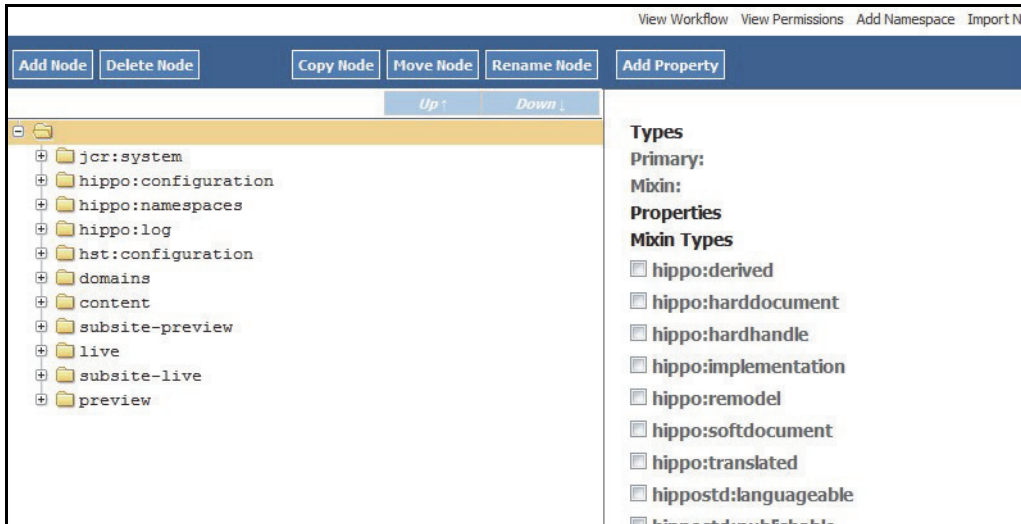


Figure 198. The console allows you to traverse the Jackrabbit repository and add or modify nodes from your browser.

However, as with Adobe and Magnolia (the other two WCM platforms using some version of Jackrabbit), this repository is another new framework for your developers to learn — one that often brings a steep learning curve.

Hippo is a decoupled, “frying” system, with separate CMS and “Delivery Tier” environments. These share the same repository, although Version 7.9 introduced the optional setup of deploying distinct repositories that you sync together. Test this, since it’s new. Both the CMS and delivery tier run in n-tier architectures, and some larger customers have successfully scaled the platform for high volumes of traffic.

The CMS itself is essentially a web application providing an editorial UI into the repository based on the Apache Wicket web application framework. Fortunately, this is an industry standard UI framework so you can modify the editorial UI (and many clients do). On the down side, it’s a different framework from what’s used on the delivery tier, so it’s another subsystem to learn. The company says it’s working on transitioning to Angular JS, instead.

The Hippo delivery tier is essentially a front-end development platform for streaming content from the repository via JSP or Freemarker, using tag libraries. It ships with a set of developer tools for building websites using modular, component-based interaction and resource processing, discussed more in “Development” on page 412 of this evaluation.

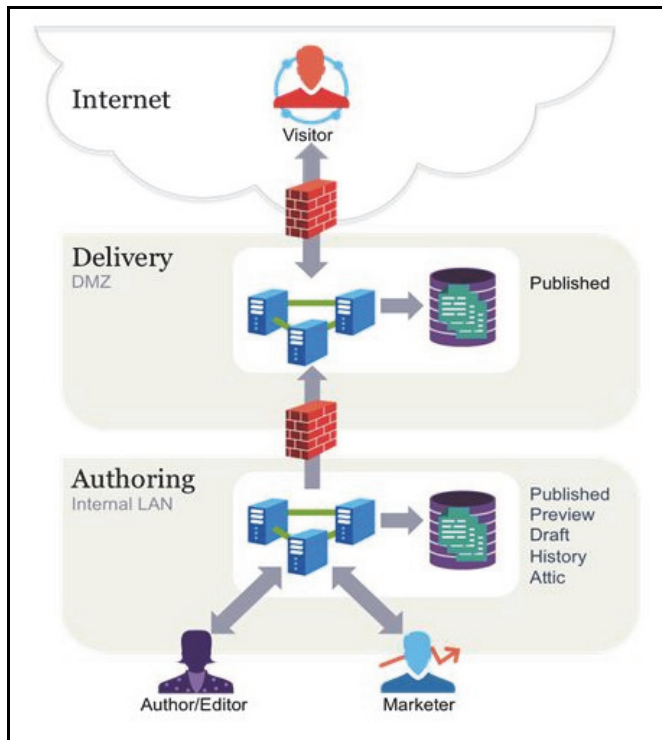


Figure 199. A typical deployment of Hippo CMS includes the JCR and the CMS sitting on a Tomcat web server in authoring and delivery environments. Source: Hippo.

With respect to the cloud, Hippo can provide managed hosting itself or you can deploy the platform within Amazon. Popular Amazon service providers like Bitnami have not yet taken up Hippo; thus, you don't have as many cloud reseller options as you might with Drupal, for example.

Hippo CMS puts users in groups, which then have "roles" within different parts of the repository (including the sections of the repository where various configurations are stored, so you can set permissions on those, too). However, as with many aspects of the system, the underlying mechanism is more powerful than the interface. You may have to apply some of your access configurations directly in the repository, rather than use the administrative UI.

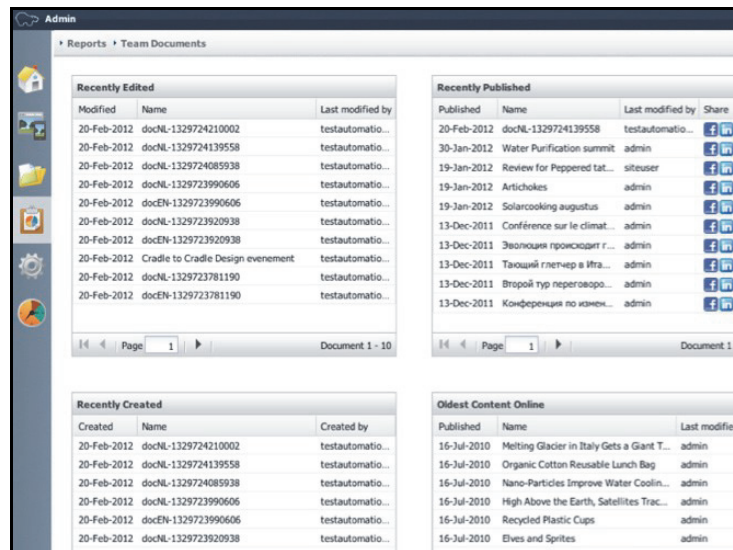


Figure 200. Workflow reporting in Hippo CMS provides a quick view into recent activity.

System reporting is comparatively comprehensive, especially for content-oriented metrics such as workflow reporting, broken links, document totals, and recent activity per user.

Development

Hippo Site Toolkit 2 is the main developer tool in the Hippo CMS environment. It is a templating engine that is exposed in the Hippo CMS UI, and it uses a tag library and expression language to develop JSP- and Freemarker-based pages.

In theory, HST2 provides a mechanism for clear separation of business logic and behavior (in Java HST components) from content (managed in the repository) and presentation (the JSPs). In practice, you need some advanced skills to achieve this.

The HST2 Tag Libraries are split in “modules” that can be modified to purpose. The key elements reside in the repository, so you control templates, configuration, URL patterns, components, page definitions, targeting rules, and more directly from the administrative UI. (The actual website application is generated as a WAR file that lives on the file system.)

You can use the administrative UI to define sites, templates, page definitions, and other parameters. This is somewhat more convenient than using the JCR browser, but despite well-meaning explanations in the dialogs, it remains a fairly technical exercise. In its current form, it’s unlikely to be useful for anybody that was unable to perform these tasks in the JCR browser in the first place.

Hippo’s HST2 also serves as a JSP delivery tier. The default installation of Hippo ships with separate preview and live environments. HST2 provides a mechanism for clear separation of business logic and behavior (in Java HST components) from content (managed in the repository) and presentation (the JSPs).

Developers sometimes complain that Hippo has a dearth of cohesiveness. Despite the introduction of Channel Manager in version 7.7 as well as development console improvements in the “developer love” version 7.6, there are still many areas that drive coders crazy while developing in Hippo CMS.

One of these areas where developers often get confused — especially in their early stages of working with the CMS — is the terminology. The term `hst:page`, for example, doesn’t refer to the actual web page. The page is what hides behind the combination of a template for an `hst:page` that refers to an `hst:template`. However at the same time, both `hst:page` and `hst:template` can act as templating engines to control layout of a page.

On the plus side, Hippo has added facilities to package up and transport packages of content and code via zip files across environments. This is handy for configuration management and general testing.

Performance

Generally, customers do not encounter performance issues on relatively small-scale sites, if caching and scaling has been configured properly. Test the Hippo CMS for performance and scaling carefully if you’re dealing with large websites or high-demand environments.

Hippo has a history of early CMS builds that suffered badly from inefficient connections to the Java Content Repository. This has been swapped out for a more tightly bound connection, which has been a noticeable improvement; however, you’ll still want to test. (Some larger implementations report decent experiences with clustering or by scaling out the various components horizontally and vertically.)

Content

Contributor Experience

The interface of Hippo CMS has been modernized throughout its many iterations, and now features a work-area-based layout. The layout is feature-rich, albeit a bit busy. In the left-hand pane is the content tree, where you can browse documents and images, which you process in the main, right-hand pane.

On the whole, most content editors report that it's relatively easy to find your way through the system once you get to know it.

The use of Wicket allows for a fluid user experience, and the drag-and-drop functionality certainly adds to usability.

However, when working on content items themselves, editors may struggle with a rather sparse experience that seems to have been designed more for gadgeteers and techies, as opposed to the more marketing-oriented, editorial community.

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

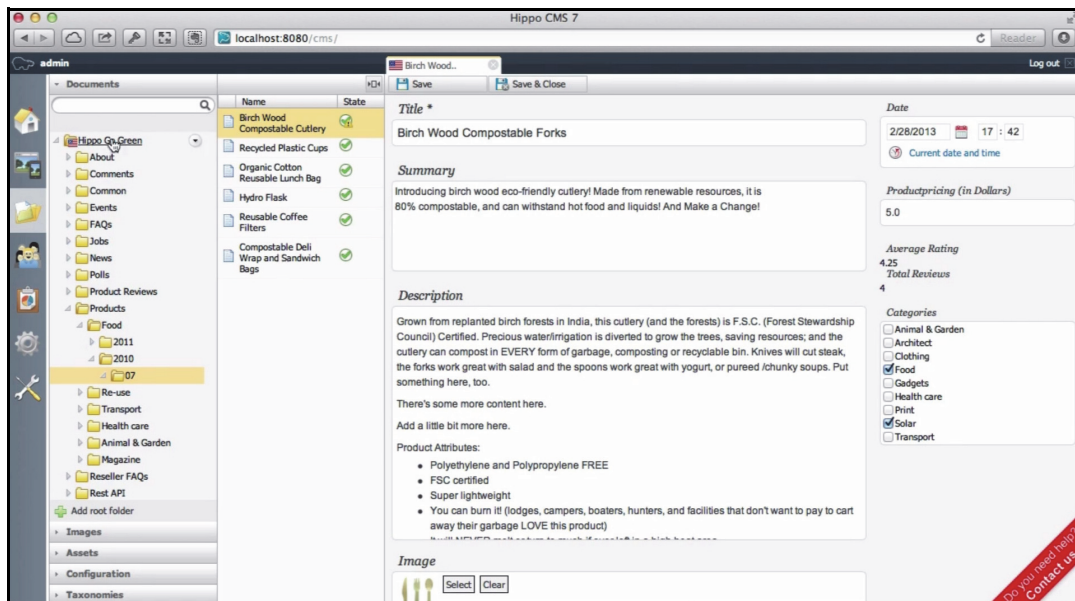


Figure 201. Hippo's default management interface has morphed over the years to five column-panes. It's highly functional, but you'll want to have high-resolution settings to exploit it.

Hippo's "regular" interface feels a bit dated. For instance, to create a new item, a Windows user would expect to be able to right click in the list of items to "create new" (something that SDL already understood years ago). Instead, you have to click on the parent folder in the left hand pane, which opens a pull-down menu for these functions. It is a UI metaphor that isn't

native to any OS, and one of the many minor quirks to which your editors will need to get accustomed.

In general, the overall contributor experience isn't a prohibitive problem, but the dialogs are too technical for most users to comprehend, and the minor bugs also detract from the experience. One customer told us that they considered the Hippo UX "hideously ugly, like an erector set construction." That may be an overstatement, but it's indicative of a general lack of customer love.

Do hands-on testing with a current version before committing. If your content editors do not grasp how to work in the Hippo CMS rather quickly, chances are that they will continue to struggle throughout the lifetime of the Hippo CMS implementation.

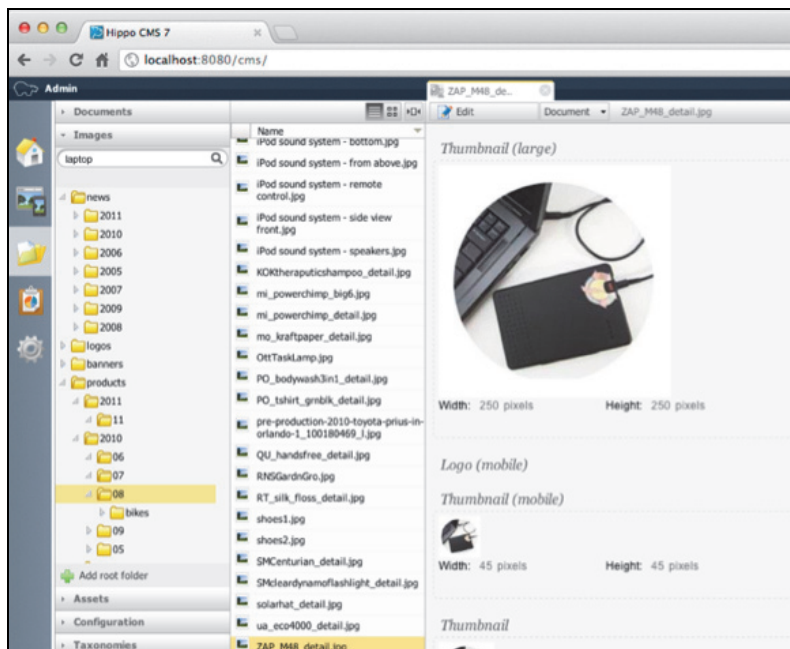


Figure 202. You can set up image scaling options in Hippo and specify what images/dimensions are appropriate for each channel.

Contributing Content

At its core, Hippo is really a content component publishing system. You publish discrete pieces of content, which are assigned to channels explicitly or based on metadata. It's not a page-oriented system; rather it is more Drupal-like in the way that content lives independently from structure.

When opening the interface, the left-hand pane is divided in the content tree, and a stack on the bottom pulls up documents or images. The top menu is now reserved for direct actions in the CMS only (such as "add image"). On the far left, buttons point to the main work areas:

- A dashboard, with an overview of recent activity and tasks in the workflow to complete
- Documents, all editable content
- The control panel for administrative tasks

The right pane will open content items to be edited. Hippo CMS uses the CK rich text editor (new in version 7.9). It's possible to open multiple tabs there, so you can easily switch back and forth between items — very handy.

Recent versions have seen Hippo apply more widget-like functionality to the editorial experience. You can drag components onto pages and then configure them to hold the relevant content, and this page-building catches Hippo up to some commercial systems. In addition, you need to be careful about whether you are modifying the template or an individual page. For the most part, it's the former, which can be disconcerting on larger, more distributed sites. Such are the vagaries of page management within “placeless” content management systems.

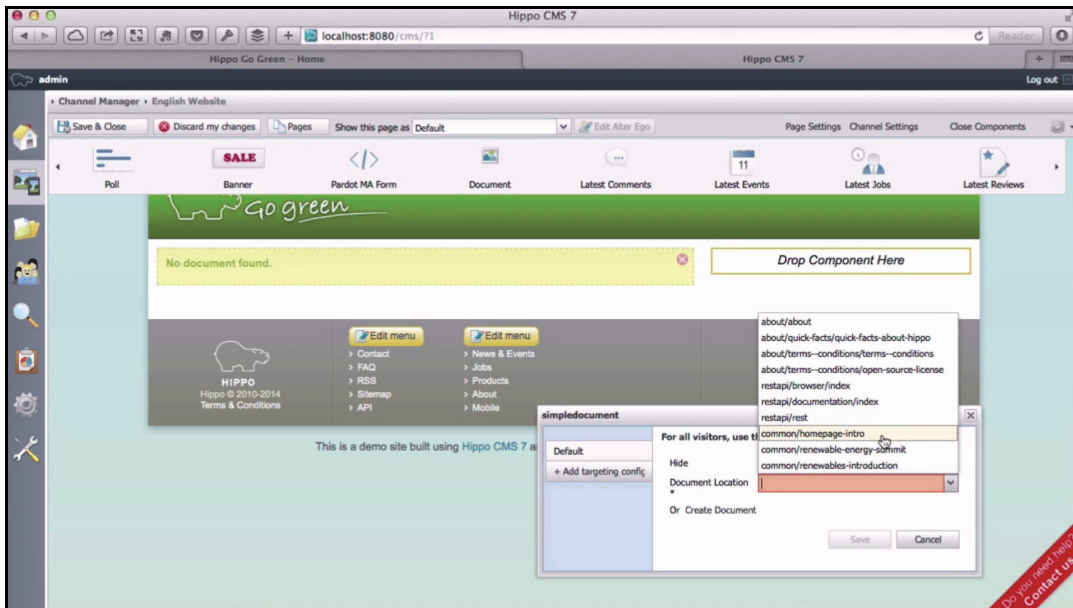


Figure 203. You can drag components onto a page and then configure them to display the right content.

Images get organized very simply in folders. The CMS provides light functionality for cropping and resizing, and renditioning; otherwise, media handling is not a strength of this platform.

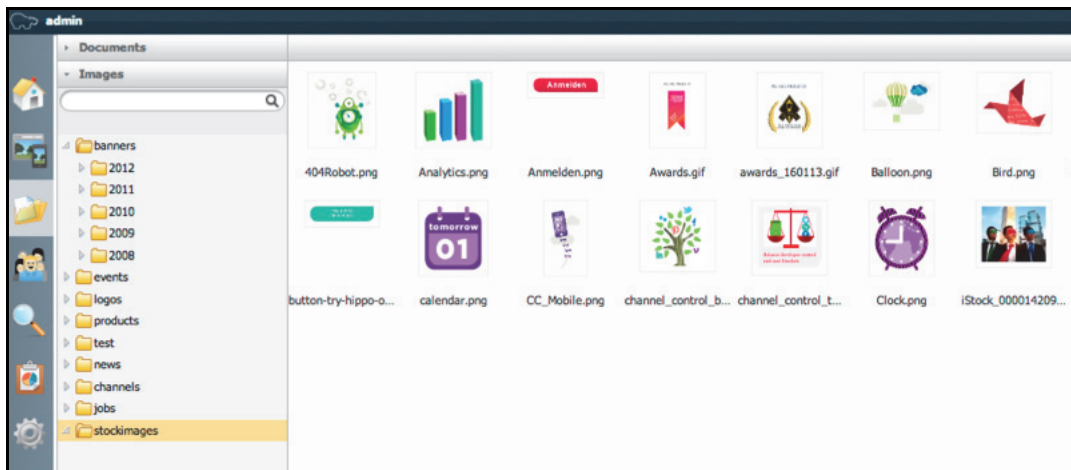


Figure 204. Hippo’s rather prosaic asset library.

Hippo seems better suited for re-purposing entire documents, rather than reusing chunks. This is not necessarily a bad thing; most customers leverage the platform for basic, Multichannel Publishing, rather than more componentized content management.

Just note that — like many systems at this tier — your contributors may struggle a bit to understand where in the editorial environment a particular article might be published, which makes multichannel publishing a slightly more fraught process.

For repository search, Hippo employs Lucene functionality, embedded in the Jackrabbit repository. It will search binary attachments if they contain text (such as Office documents and PDFs).

Content Lifecycle

Workflow in Hippo CMS gets configured from the repository. There are several basic workflow rules available out of the box, but if you want to customize them, get ready to roll up your technical sleeves; all customizations require coding (customizations or even extensions).

Hippo has improved workflow usability in recent versions with features like a to-do list in the dashboard, but some customers complain that it still feels too rigid for more collaborative-oriented web publishing efforts.

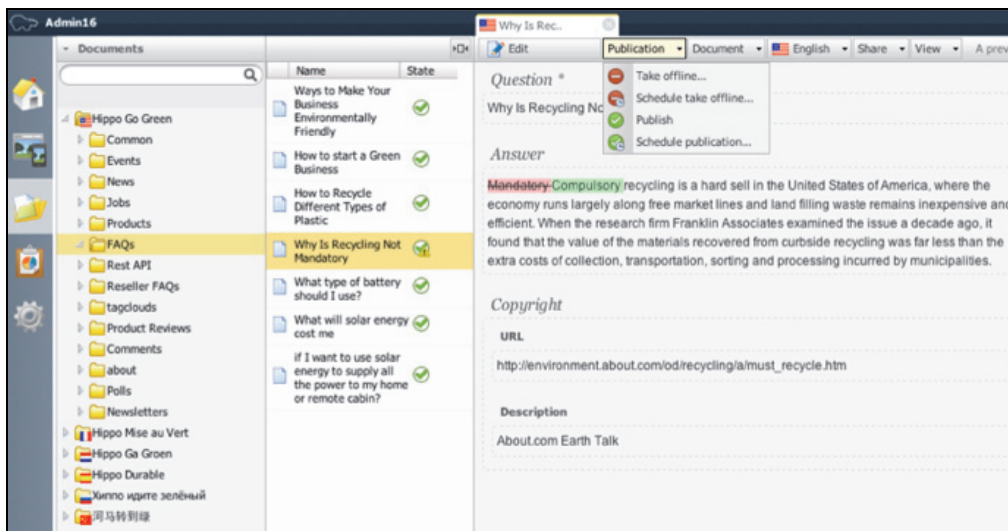


Figure 205. Workflow in Hippo CMS is exposed in the UI.

It wasn't until recent versions that Hippo added the ability to perform batch workflows. If your processes require publishing and unpublishing content from a particular folder in batches, test this functionality carefully.

Starting in version 7.5, Hippo CMS introduced a modest set of capabilities for performing translation management. For a company with European heritage, it is rather disappointing to see that translation and multilingual capabilities were not addressed earlier in the product's lifecycle. What you currently get is the ability to translate your website structure, content, and URLs into different languages.

Once you manually create translatable versions, they can be navigated in the repository. Note that all source and target language versions are currently stored at the same level without any

parent-child relationships. The only inheritable element is metadata, which is a definite limitation on efficiently reusing content.

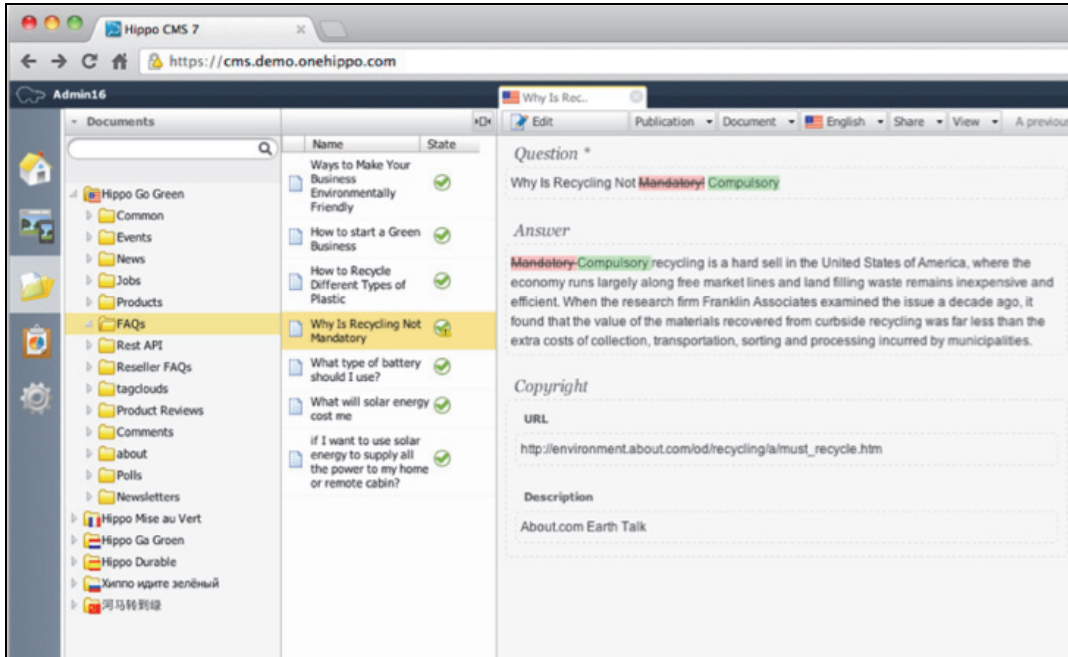


Figure 206. The version compare feature in Hippo CMS shows changes highlighted in different colors with the ability to roll back to older versions.

Experience

Publishing

You can configure Hippo to output relatively friendly, standards-compliant code, with friendly file names.

As an XML-based system, it lends itself readily to multichannel and mobile delivery, although you may find specialized services are missing; it's really a development platform with good promise.

Hippo did not start paying serious attention to multisite and multichannel management until version 7.5, when they released Channel Manager. The Channel Manager's purpose is to manage content across various channels: from web, to mobile, to social networks.

Hippo refers to each website as a "Channel." As with other systems in the WCM market, you can define different target devices and specify which

Experience	
Publishing	
Standards Adherence	●
Multichannel	●
Mobile	●
Digital Marketing	
Site & Campaign Analytics	●
Testing & Optimization	●
Segmentation & Personalization	●
Social Media Integration	●
Promotional Campaigns	●
Community & UCG	●
Workplace	
Collaboration & Networking	○
Dashboard	○
Ancillary	
Site Search	●
Online Forms	●
Module Ecosystem	●

templates and layouts should be used for each one. A “channel” can be a website in a different language, a different application, or a mobile device.

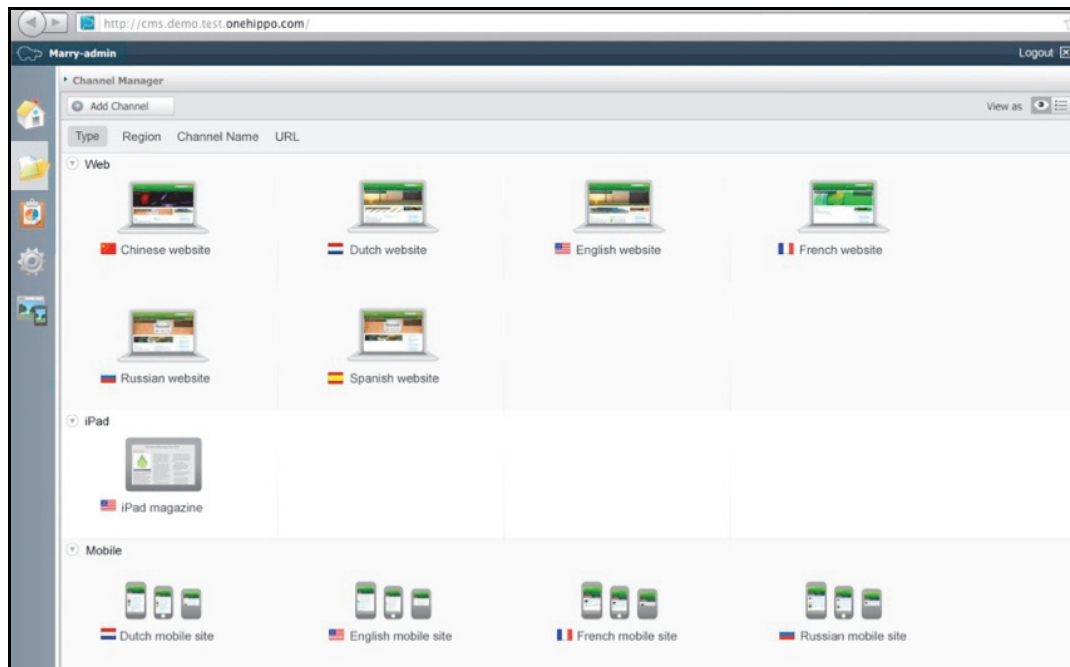


Figure 207. Hippo’s Channel Manager from initial setup to being able to preview content on different channels within the CMS interface.

When you need to add a new channel, you will have to go into the Blueprint — a name familiar to those of you who use a CMS developed in a neighboring district of Amsterdam by SDL. Conceptually, it makes sense and works out of the box. One of the main challenges with this tool will be your ability to reuse functionality, code, and content models across a broader deployment featuring several websites.

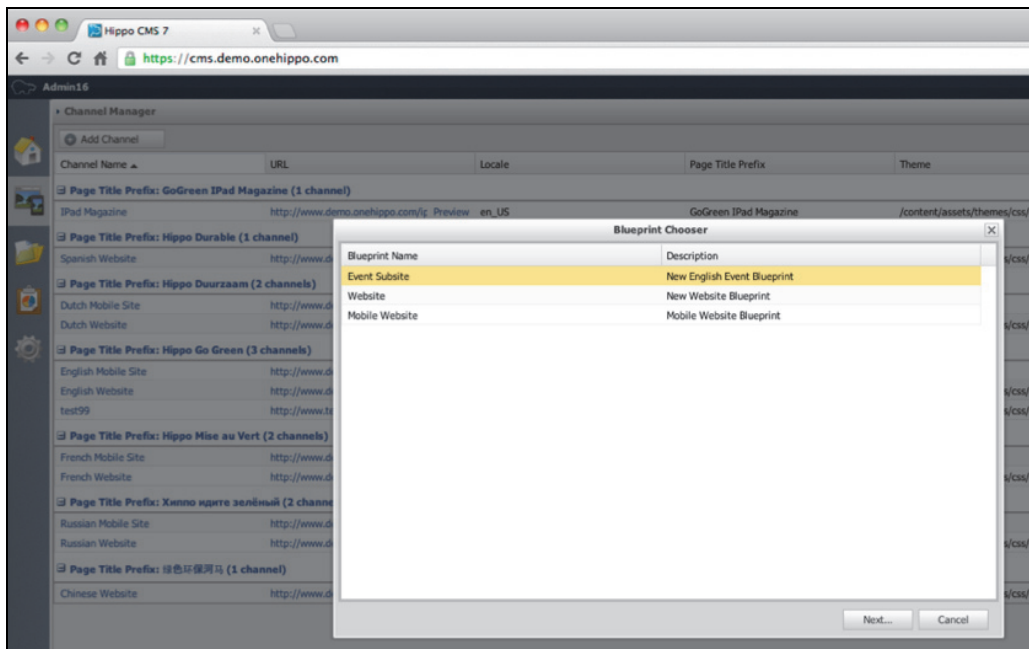


Figure 208. Adding a new channel in Hippo's Blueprinting model.

Digital Marketing

In recent editions, Hippo started adding more interesting personalization and reporting capabilities. For example, the Template Composer functionality can be used, among other tasks, for A/B testing.

Version 7.9 in particular amped up the platform's personalization capabilities. You can create composite personas based on a variety of attributes — including daypart, locale, visitor behavior, and so on — and then assign those categories to variable content in the editorial process. The tasking here feels a bit clunky, but it works. You can then preview by impersonating those personas in the in-context editorial environment via a pull-down menu.

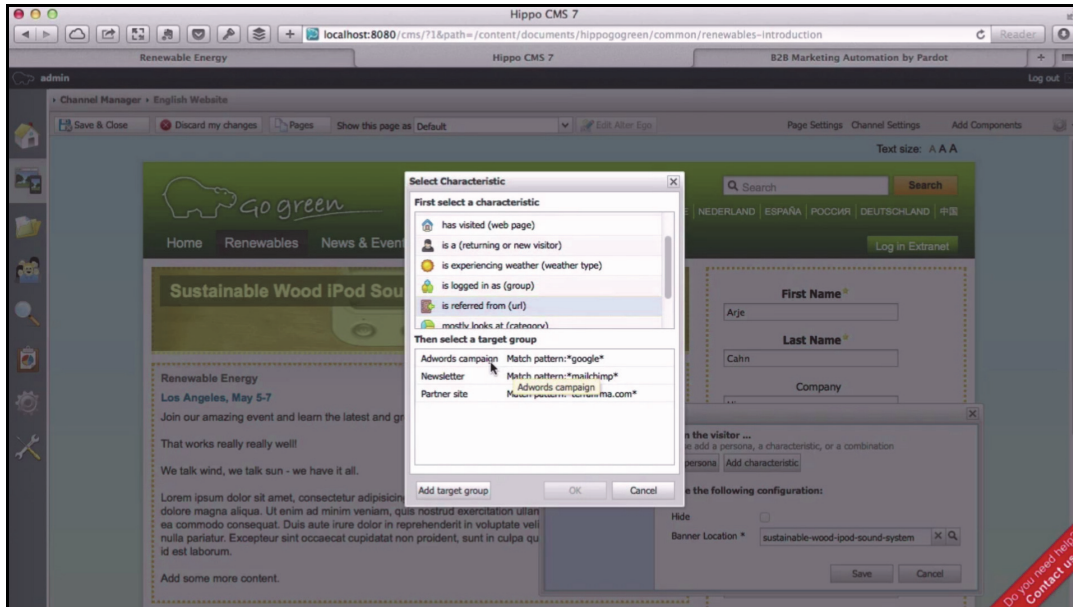


Figure 209. Building user characteristics in Hippo.

These new features are quite promising, but few Hippo customers have genuinely exploited them in the field yet. You'll want to test carefully—in terms of usability and performance (since personalization is always the enemy of caching).

On the social media integration side, you will not find much power or flexibility in Hippo CMS. In theory, you could potentially use Channel Manager for Facebook page management, but so far, it seems impractical.

Hippo plays well with third-party marketing automation and email campaign tools. A new landing page creator wizard helps you build destination pages that you can integrate with Pardot, Eloqua, and Marketo, among others. This is new; be sure to test carefully.

Hippo's Google Analytics integration provides basic views into statistical information for your website. Hippo has invested some effort into creating useful dashboards based on Google Analytics data, but they are only useful after you carefully set your configurations on both platforms.

On other hand, any analytics reporting in Hippo CMS will require additional development efforts if you choose to integrate with Google Analytics or any other web analytics system like Omniture or Webtrends. This feels like an odd omission.

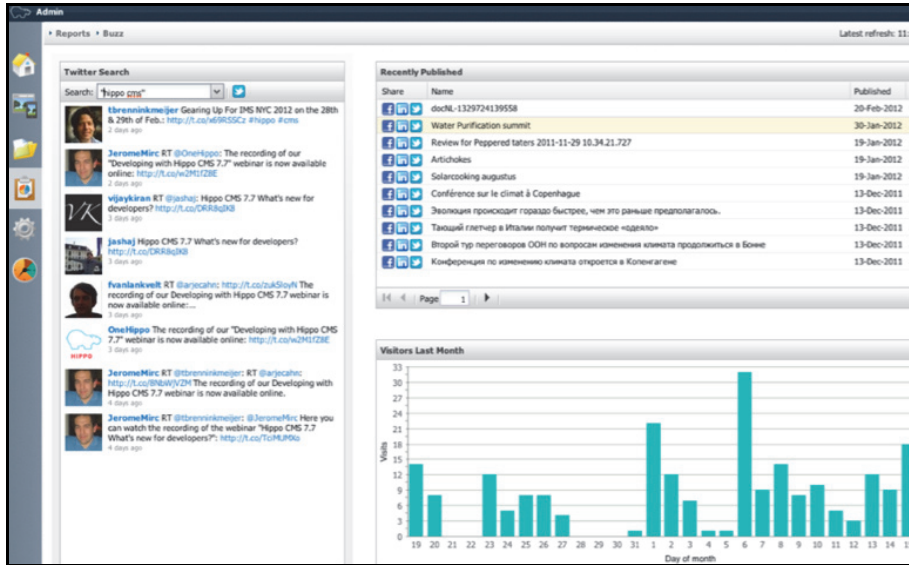


Figure 210. Social analytics reporting via internal queries and Google Analytics integration provides some minimalistic insight into social intelligence within the Hippo CMS admin UI.

Ancillary Services

Site search (and repository search) uses Lucene Solr, and includes capabilities to allow you to search Office documents and PDFs out of the box.

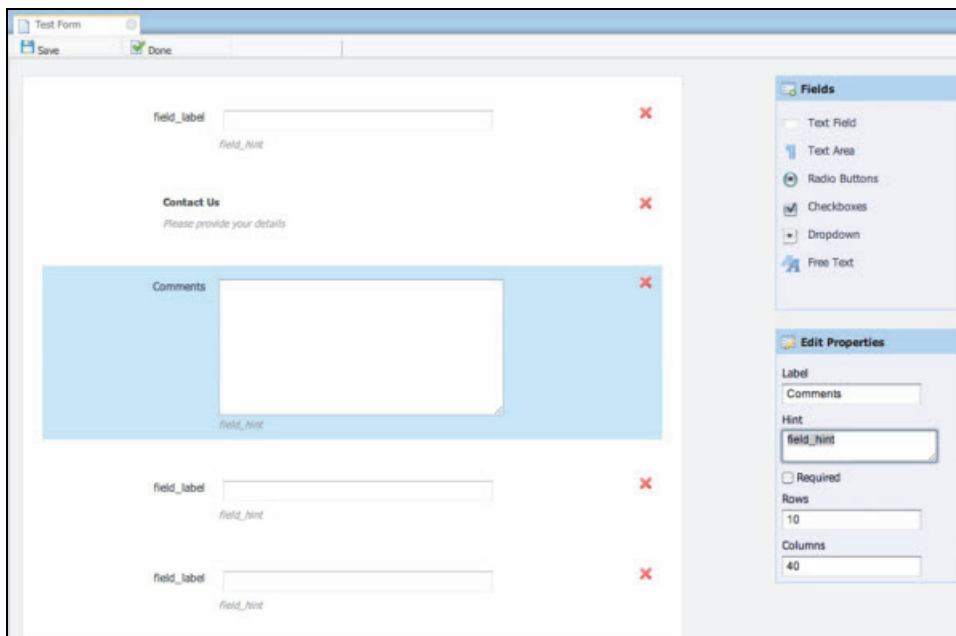


Figure 211. Hippo web forms.

The HST2 framework offers simple blogging and commenting services, but don't expect anything fancy here.

As with many other products in the Web CMS market, Hippo CMS has the ability to build web forms within the content management system.

Web forms are treated like any other content type in Hippo; they are stored in the repository, which means they can be published, version controlled and reused across different pages. You may want to leave web forms in the hands of your technical personnel, since content editors and marketers are likely to find Hippo’s techie approach too intimidating for web form building.

Vendor Intangibles

Hippo started as an open source project, but Hippo the company sells commercial support and consulting. The company still maintains — and even dominates — the codebase. Of the 25 committers, 20 work for Hippo B.V. The company also holds all copyrights, but distributes the software under a relatively loose, Apache-style license.

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

However, understand that (unlike Plone or TYPO3) there is no formal community foundation or governance mechanism beyond Hippo B.V.’s leadership. Thus, you are really tying your cart to their horse, regardless of how vibrant the community becomes.

Hippo itself makes money by selling support, training, and — above all — licenses. According to Hippo, the most recent growth has come from North America, where Hippo has acquired several key customers, which are now served from the Boston office.

A commercial license costs start at \$50K, while median implementations cost at least twice the license amount. Hippo charges support per CPU, per year. The median annual fee runs about \$50,000, but can go up from there.

Although Hippo the software is a very developer-intensive platform, it does not appear to have attracted integrator and consultancy enthusiasm the way Drupal has. It could be that Hippo B.V.’s comprehensive support offering and dominant profile in the community pushes away would-be competitors. Most of the optional modules — while freely available — were developed by Hippo itself. Much of the community activity beyond Hippo B.V. seems to be focused on end-user customers that dip in and out as their projects evolve.

Indeed, even in Benelux, customers say it can be hard to find experienced implementation support. Some partners seem to think that Hippo B.V. itself will help with implementations, but over time, Hippo has tended to shy away from consulting in favor of support and product development. In general, this is a wise business choice, although it can sometimes leave smaller integrators exposed, and you’ll want to test the project management skills of any implementer — including Hippo itself. However, paying customers do laud the company’s product support, which is a rarity in this marketplace.

The company says it is working on expanding its partner channel, especially in North America. If successful, it bodes well, but building consultancy expertise on a relatively complicated product will take time.

Historically, the documentation has been chaotic. You may find many outdated pages, and a plethora of broken links. Like other open source tools, Hippo supports a public mailing list, where Hippo developers actively participate and answer queries.

While Hippo usually pushes hard with new releases and actually delivers most of its promised functionality, it still can prove rough around the edges. Technical enthusiasm is not always matched by careful release cycles.

Conclusion

In the hands of an experienced Java developer, Hippo offers the promise of creating custom content applications for structured information, particularly content destined for multiple channels or devices. Few other open source tools have targeted this niche (most focusing instead on managing unstructured HTML content).

Hippo shows promise for global, multisite management, though its features do not yet match those of others (e.g., its fellow Dutch vendor, SDL Tridion). Hippo B.V. the company is technically oriented and developer friendly. Unfortunately, you may find yourself mostly on your own with your Hippo implementation due to the traditional dearth of experienced integrators (in addition, you may encounter newer and less experienced partners). You can count on Hippo to offer the initial training for your developer team, but Hippo is not strong on the professional services side. If you take on Hippo, make sure you have ample internal resources.

Finally, customers note that Hippo has a tendency to develop first and ask questions (and test) later. In recent years, this has made the platform comparatively quite innovative. When you consider that Version 7.0 was essentially a complete re-write of an older platform, the technology has come a long way in nine dot-releases since that time. Still, customers with lower risks tolerances may find Hippo a bit loose.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Microsoft: SharePoint Server 2013

sharepoint.microsoft.com

Vendor at a Glance

Specsheet	Microsoft: SharePoint Server 2013 Summary
Geography	Global
What's New	<ul style="list-style-type: none"> • SharePoint 2013 boasts major new enhancements and better pricing • Heavy-duty field experience with the new platform remains limited
Strengths	<ul style="list-style-type: none"> • Built on top of core SharePoint repository and library services, so you can leverage some SharePoint expertise for web publishing • Broad development platform to create content-oriented web apps • Microsoft has improved SharePoint's content contribution services and the system defaults to more standardized markup • Includes a full license to the powerful FAST Search engine • Platform boasts exceptionally broad, unofficial support community
Weaknesses	<ul style="list-style-type: none"> • Requires enterprise web publishing teams to do things "SharePoint's way," with careful, up-front configuration planning • Very weak digital marketing features • Lack of packaged components means WCM efforts become development projects more quickly than collaboration scenarios • Surprisingly buggy and unexpected behavior, compared to more solid, previous releases • More onerous hardware requirements, comparatively pricey licensing, and high customization needs continue to make SharePoint 2013 a poor value for WCXM scenarios • Three-year upgrade schedule is a mismatch for fast-moving innovation cycles in this space
Potential Fit	Community-Oriented Site, SMB/Departmental Digital Workplace
Unlikely Fit	All Complex website scenarios
Compare To	Ektron, Sitecore, EPiServer, Telerik, Kentico
Operating System	Microsoft Windows Server 2008
Repository	Database: MS SQL Server 2008
Client	All major browsers supported
App Platform	.NET
Licensing	Server and seats
Ownership	Public (NASDAQ: MSFT)

Summary

Microsoft SharePoint 2013 represents a decent upgrade from SharePoint 2010, but it's not a game-changer in this fast-moving space. While Microsoft has made many improvements in the areas of content authoring, taxonomy, reuse and usability, SharePoint still remains more of a development platform — especially when it comes to large public-facing websites.

In short, the fundamentals of web publishing SharePoint 2013 have not changed significantly from 2010; therefore, the platform remains more developer-focused than marketer-oriented.

The vast Microsoft ecosystem will come to learn about and build upon the new platform, but many (if not most) consultancies have eschewed SharePoint for web publishing scenarios, particularly for public sites.

Savvier system integrators (those with experience across multiple CMS tools) have traditionally reported much higher development work and longer schedules with SharePoint implementations. While many will recommend SharePoint for internally facing or collaboration-oriented intranet zones, some parts of Microsoft actually will recommend other WCM-specific tools before they recommend SharePoint — especially for public sites. We don't believe this will change with the 2013 edition.

SharePoint is best suited to Digital Workplace scenarios, although probably not ones with lots of managed HTML pages (as opposed to collaborative documents, where SharePoint does much better). For other scenarios, it is really a “tweener:” too complicated and development-heavy for basic sites, but too clumsy and non-marketing-oriented for larger sites.

Scenario Fits	
Simpler Site	
Informational	<input type="radio"/>
Microsites & Landing	<input type="radio"/>
Mid-Range	
Basic Digital Marketing	<input type="radio"/>
Mobile Site	<input type="radio"/>
Community Oriented	<input checked="" type="radio"/>
Complex Site	
Advanced Marketing Portal	<input type="radio"/>
Global Enterprise	<input type="radio"/>
Multichannel Publishing	<input type="radio"/>
Ultra-Large Single	<input type="radio"/>
Digital Workplace	
SMB/Departmental	<input checked="" type="radio"/>
Global/Enterprise	<input checked="" type="radio"/>

Introduction

First, a little history: In 2001 Microsoft acquired a Windows-centric CMS package, NCompass Resolution. The package was renamed and re-launched as Microsoft Content Management Server (MCMS). Over the next few years, Redmond put only limited effort into the product and it was wholly replaced by SharePoint 2007 WCM.

For WCM, the SharePoint platform has been playing catch-up ever since. With SharePoint 2010 came some much-ballyhooed — but ultimately quite modest — improvements for web publishing. The same could be said for SharePoint 2013, released to GA in late 2012 after an extended beta period.

For the purposes of this review, we focus on the enterprise edition of SharePoint 2013. The standard CAL misses many key features, and the cloud-based, Office 365 edition isn't really meant for enterprise web publishing at all.

To be clear, our focus here is on SharePoint as a web publishing platform. We do not go into great detail about SharePoint’s many other capabilities, which are covered in detail in our different research streams.

Technology

Technical Administration and Security

A SharePoint environment is actually made up of several different Microsoft systems. Base systems underneath SharePoint include:

- Windows Server
- Active Directory
- .NET Framework
- IIS & SQL Server

Note that the installation has gotten a bit more complicated this time around, with a number of additional SQL Server and .NET modules required on each server. SharePoint is I/O-hungry; you need lots of RAM (ideally 16GB) and at least 4 cores (ideally 8+).

SharePoint 2013 is built upon (the free) SharePoint Foundation, but you would never want to use Foundation alone for web publishing. The platform takes advantage of some fairly mature SharePoint services around version control, security, search, Web Parts (Microsoft-speak for portlets), and more.

Note that SharePoint still stores files natively in SQL Server. This makes them potentially more conducive to indexing, but storing files in a relational database can affect performance. Remember that there is a limit of 2 GB size per file.

A SharePoint installation starts with the notions of “sites,” which can be rolled up into “site collections,” which themselves can be aggregated into “applications” within a “farm” (single instance of SharePoint), where various other “shared services” may reside. Sites can be made up of “lists” and “libraries,” which aggregate different information elements, which Microsoft calls “columns.” “Content types” are another, more abstract way of combining columns into an organizational unit. Individual web pages are typically comprised of one or (usually) more “Web Parts,” which connect the user to underlying content and services.

In SharePoint, when you want to create a traditional website (as opposed to a collaboration or BI site), you explicitly set it as a “Publishing” site. This is not activated by default, and if you forget to activate for Publishing before creating a publishable site, you may be greeted by useless messages.

Like SP 2010, SharePoint 2013 has a coupled production and delivery architecture; you use the same software to produce content as well as to deliver it. As always, this simplifies deployment, but it can mitigate against a cleaner separation of tiers. This also eliminates the possibility of SharePoint deploying a “flat” HTML version of the site to any remote web server, which is possible using other WCM products.

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

However, SP 2013 brings some new twists here with an updated architectural approach called “Cross-Site Publishing.” The name implies a solution to the old SharePoint problem of the inability to share content across site boundaries easily; in fact — for better or worse — it is much more than that. Using this approach, you separate authoring and publishing into different collections; content authored goes into an indexable “catalog,” and then you use FAST to index and deliver dynamic content on a loosely coupled front-end..

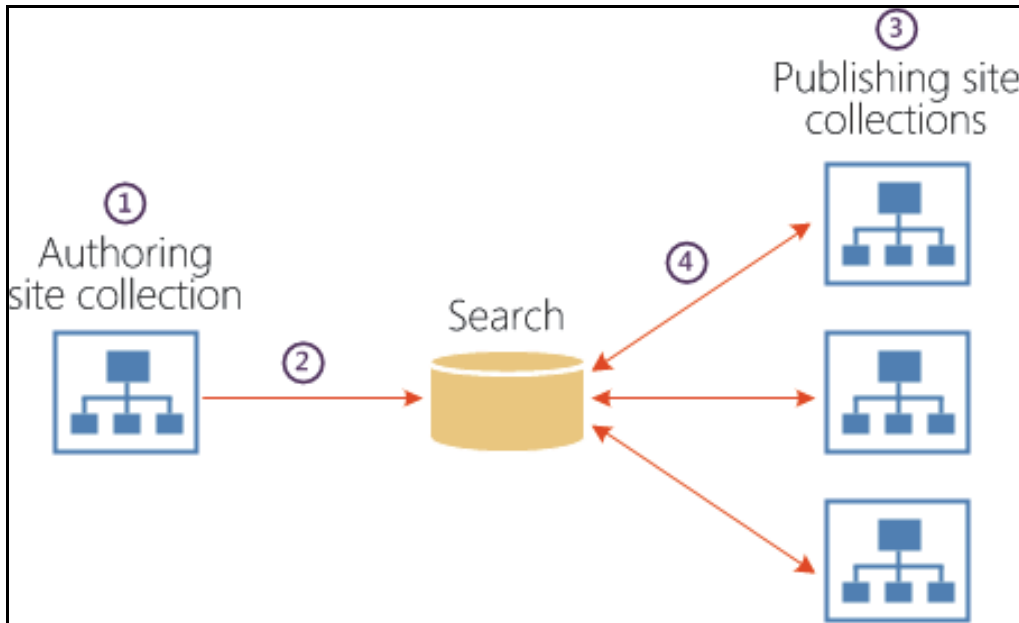


Figure 212. Simplified architecture for Cross-Site Publishing.
 Source: Microsoft.

It’s important to understand how Cross-Site Publishing works, because — although it’s generally optional — it’s definitely required for certain new approaches you may see in SP 2013 demos for services like personalization, localization, metadata-driven topic pages (like you find in Drupal), and more. The reference case for this is a genuine product catalog in an e-retailing environment, but Microsoft wants to extend the paradigm for nearly all dynamic content more generally. Note that Cross-Site Publishing is not available in SharePoint Online.

Here’s how it works. First, you designate a list or library as a “catalog.” FAST then indexes this content and makes it available to “Publishing Site Collections” via a new “Content Search Web Part” (CSWP), which is a new alternative to the old — and quite frustrating — Content Query Web Part (CQWP). Microsoft has put some good thought into creating and customizing CSWP instances, including some browser-based configurations. In theory, runtime queries should execute faster against the FAST index than against a SharePoint database.

Note that there are several downsides to this approach:

1. Serving dynamic content via search index has been controversial in the CMS world, mostly because it can require advanced knowledge of your search subsystems and associated security, metadata, and performance settings, and it engenders greater unpredictability with various moving pieces.

2. As just one example, there are bugs and inconsistencies in the FAST indexing process, such as overriding explicit metadata with implicit values, which leads to unexpected results in the presentation tier.
3. There is no readily apparent way to override or reorder dynamic lists in the delivery environment, which is an increasingly important content curatorial feature.
4. Preview, testing, publishing, and caching can become a fraught process due to various latencies. By default, the FAST crawler runs every 15 minutes. Some customers have bumped this to every minute, but that will surely have performance implications on larger collections. You cannot push content to FAST; it needs to come to you. There is no “immediate publish” in this scenario; to be fair, this can present a challenge in competing systems that also do batch publishing jobs.
5. CWSPs are easy to create in the SharePoint reference site, Contoso, but as your own content catalog collections grow in complexity and size, so too will your confusion navigating FAST indexes to create just the right queries.
6. Binaries need to be stored in a single, separate location in the delivery tier. In other words, there’s no separation of concerns with this class of content, and no notion of publishing here. You need to wire this carefully into your authoring and delivery interfaces, with appropriate security and access controls, as well as naming and directory schemes, which should be planned in advance.
7. Like so many things in SharePoint, you need to do a lot of up-front analysis here. In particular, you need to know in advance which lists and libraries you want to designate as “catalogs,” and whether that content will get localized via “variations.”

Authentication, Authorization, and Reporting

For authentication, the product is quite flexible. In addition to Active Directory (AD), it will also support LDAP, or any other ASP.NET-supported role provider. If you do not employ AD however, your users no longer access SharePoint directly from within Office.

There are two types of Reporting in SharePoint 2013. The first kind of reporting is about monitoring the health of the infrastructure, which includes things like CPU utilization, memory usage and so on. The other kind of reporting is related to near-term visit data, which we address further below.

Development & Templating

You could think of SharePoint as a platform for building content applications in .NET. Since the product natively supports .NET and ASP.NET, you can embed other objects and services into your CMS using standard Microsoft tools and methodologies, and take advantage of specific .NET services. This is not simple, however, and the novice .NET developer can get (you) into deep trouble — very quickly.

Developers will be pleased to use a familiar tool, Visual Studio. However, there are quite a few additional tools that you will need to use for various development needs. There are certain areas like workflow and templating where Redmond urges you to employ SharePoint Designer.

The good news is that you do not need to have your development environment on a server machine; most developers run SharePoint right off a laptop. Moreover, developers have access to an ever-wider set of starter components and templates from which to build.

Notions of templating begin with a “Master Page,” an ASP.NET construct that defines the basic page structure — headers and footers, navigation, logos, search box, and so on — essentially the chrome of your site. Traditionally, this is where things got tricky in SharePoint, because master pages tended to contain a lot of cruft by default, and “branding” a SharePoint publishing site was somewhat of a dark art. Beneath master pages come “Page Layouts,” essentially sub-templates, which are the fields and ordering of components within the body of the master pages.

In SP 2013, Redmond has certainly improved things, most notably via a new “Design Manager” module, which is essentially a WYSIWYG master page / page layout builder. Design Manager is essentially an ASP.NET and JavaScript code generator. You upload HTML and CSS files that you create and preview offline. After you add additional components in the UI (like specialized Web Parts), Design Manager generates the associated master page. Page layouts get converted to SharePoint-specific JavaScript that the platform uses to render the dynamic components on the page. Want to update that master page? Just upload an updated version of your HTML/CSS file and rerun the generator.

You can generate and propagate a “design package” to reuse designs across site collections. That’s handy, though not as object-oriented as other, more nested templating subsystems you find in nearly all other products at this tier. There is a notion of template snippets that enable you to apply layouts within a design package, but they are not reusable across design packages.

This process is certainly more straightforward than the previous approach, but note that it still would likely involve a developer, and could prove clumsy when making small tweaks to existing designs. Moreover, it still tends to commingle business logic with display logic, which makes it harder to separate concerns (and skills).

Performance

As a fairly traditional portal offering, SharePoint comes with various approaches here, including adding more machines at different tiers, and offloading various services (such as Search) to their own servers. Performance is always an issue with any Microsoft server product, but you can address it if you are willing to invest in more servers. It looks like Redmond generally is recommending more specialized, service-specific servers for SP 2013, so you’ll want to plan for more iron.

As with most other platforms, caching can often become a complex subsystem; for high-traffic, public sites, test it carefully. With SP 2013 though, Microsoft added some useful tiers to SharePoint, including object, BLOB, page caching, and anonymous search caching. You have access to unusually rich interfaces to configure and monitor these caches.

Unfortunately, cache invalidation (“flushing”) rules seem more opaque, and this could create some extra help desk calls. The system seems to rely more on full-cache flushing than other products, which could lead to unexpected performance troughs.

Note that SharePoint still stores files natively in SQL Server. This makes them potentially more conducive to indexing, but it can slow performance.

Content

Contributor Experience

For content contributors, the Office ribbon from SP 2010 has improved, is a bit more streamlined, and is also more readily customizable — a good thing. The SP 2013 interface could still confuse less technical users, but for the more technically inclined administrator or power user, many controls are just two clicks away.

SharePoint for WCXM has traditionally suffered from a comparatively poor contributor experience. This was partly due to poor cross-browser support; you basically needed Internet Explorer to “enjoy” full-fidelity editing. While 2010 improved this situation, there were still IE-only features. In addition, the standard rich text editor that shipped with both 2007 and 2010 was, conservatively, two years behind most purpose-built rich text editors. As a result, many firms implemented Telerik’s RadEditor (since it was free), or simply lived with the poor experience.

For SP 2013, Microsoft has revamped its rich text editor, which seems to behave in a more standards-adherent way. The rich text editor can be used to edit content “in-context” on your site, and it enables you to edit structured and unstructured content. The editor is still a one-off service rather than a community effort like CKEditor; it’s therefore less likely to be updated regularly.

As with everything from Redmond, the user interface has been broadly localized, with Microsoft including a very impressive array of language packs.

Contributing Content

In SharePoint, web content gets stored in a library (a special “pages” library). Each page is a list item within this library. To edit an existing page, you typically go to a page in edit mode (or create a new page), and start editing (or adding content to) it.

2013 also brings a new “Cross-Site Publishing” feature that enables pushing content from one site to another, while keeping them synchronized. This new feature relies on the newly revamped FAST Search engine. This significantly improves your content reuse capabilities (as well as personalization, which we address below), but the new service may prove tricky to manage and could lead to unexpected results and behaviors (see “Technology” on page 425 for more details). In particular for editors, it could slow the publishing process and make high-fidelity preview a major challenge.

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

With SP 2013, Microsoft has figured-out how to enable contributors to add more complex, non-Web Part elements like embedded code and video that doesn't have to be based on a specific Web Part (think an object tag for a Flash element). Called "Embed Code," the feature resembles what you would see in many competing tools. Just note that if you are using Cross-Site Publishing with its search-based delivery, widget behavior may prove finicky and could lead to problems that require support.

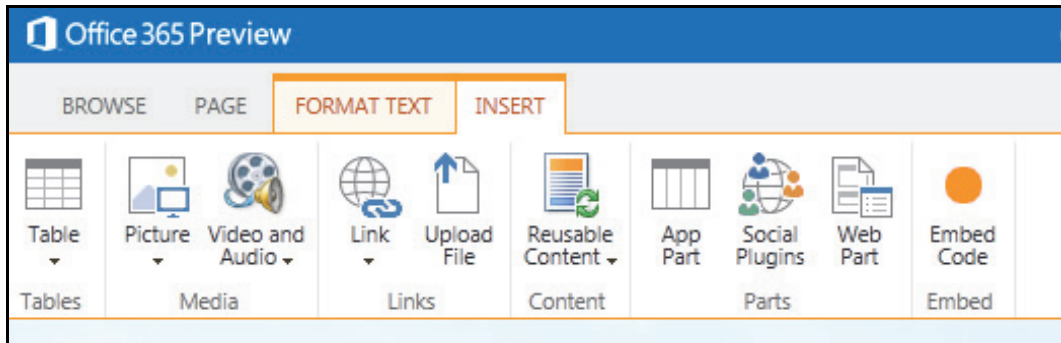


Figure 213. The Insert Ribbon for contributors.

Beyond the basic functional elements, Microsoft has also caught up to more modern application development trends with "App Parts." App Parts speak to an architectural shift that allows external web applications to integrate more directly with SharePoint without actually comingling them in one environment. Once again, you'll want to test App Parts very carefully if you are employing the snazzy new search-based dynamic delivery service.

Beyond basic content contribution, Microsoft does provide solid metadata and tagging services. The WCM features of SharePoint are inclusive of the standard metadata data and tagging capabilities that permeate the platform. In fact, natively, SharePoint ships with sample content types like the Dublin Core, which enables firms to take advantage of adding extensive descriptive and formalized metadata around web pages. Further, these content types can also include columns that source content from the term store, allowing more robust management and tagging using a standardize nomenclature.

With SP 2013, Redmond improved and simplified the term store, although you may find some more advanced taxonomy-management features still wanting. For example, there's no versioning, version control, or workflow for terms; they're not really managed content. Be careful here.

Perhaps the biggest improvement here is that using FAST, you can leverage metadata in the delivery environment much more readily than you could in previous editions, and specifically you can:

1. Employ metadata-based navigation structures (as opposed to folder hierarchies), which is a big improvement
2. Deploy automated, Drupal-like category pages and link lists based on how items are tagged

Unfortunately, getting this to work properly is tricky and it won't always produce expected results. For example, the JavaScript-based renditioning does not always behave properly. Simply adding a new page won't alter the navigation — unless you also add a new term to the term store. Moreover, at a time when other vendors follow the "content strategy" wave to

allow for post-query automated page and link curation, Microsoft has turned the whole operation over to FAST, far away from the discriminating eyes of your editors. Finally, managed navigation services are bound at the Site Collection level, and therefore must be physically copied in order to be shared across collections.

With respect to asset management SharePoint has long had the ability to store digital assets. However, once you got past uploading the FLV or PNG file, there was scant recourse to leverage it from there. Redmond has worked to improve this; for example, SP 2013 brings a new video content type, with automatic and manual thumbnailing.

Microsoft has caught up to several peers by creating “Image Renditions” capability, which allows you to contribute a full-fidelity image to a library, and then render a derivative of that image when served through a web page. This builds off some older code in SharePoint, but it is much cleaner and more integrated into the content contribution experience in SP 2013. In theory, this should better support mobile delivery as well, but like many of the newest parts of SharePoint, this service is not fully baked. Since image renditions are hard-coded into the image URL, it’s difficult have your images swap out automatically in a responsive design framework.

Content Lifecycle

One overall advantage in the way SharePoint approaches web content management is that it benefits from the core SharePoint lifecycle services. All of the core management capabilities like versioning, information management policies, check-in/out and workflow are simply standard. The challenge for you is that the user experience is still geared more toward document management than it is toward website management.

Let’s start with workflow. SharePoint comes with a default workflow system that is well-suited to the kind of simple approval processes found among web teams. Workflows are part of the core of the SharePoint foundation and can cope with general routing and approval cycle needs. In SharePoint, you create simple workflows using a browser. However, as the complexity of workflow increases, you will need to use other tools to model these workflows.

As with 2010, Microsoft addresses globalization through what it calls “variations” which allow a site to be cloned and localized. Redmond has made a lot of noise about improvements in 2013 — with some justification — but they are more of degree than kind.

With variations, SharePoint effectively copies the content from a predefined source variation into a various target “variant” sites, and tracks dependencies from there. Each variant site typically leverages the standard page layouts and master pages, but you have option of implementing custom ones. What’s new in SP 2013 is that Microsoft added a translation framework that supports either machine translation (unlikely in most cases) or exported content items singly or in bulk to a human-based service. Those translation packages can then get imported back into SharePoint.

This process works relatively well for new content and when you want to translate content outside of SharePoint. It doesn’t do a good job of isolating updated content such that you could use SharePoint as the translation environment itself.

Moreover, the relationship between sources and targets is static and could be a big drawback to SharePoint’s multilingual support. In many organizations, content may appear from affiliates in a language other than the source language. In Canada for example, new content may appear first in either English or French. Unfortunately for SharePoint users, the enterprise will have to

decide which language will be source, create content in that language first, and then translate to the target languages after the fact. This approach may work for some enterprises, but could still represent a poor fit for a truly global organization.

Experience

Publishing

SharePoint has historically struggled to align with industry standards here. For example, SP 2010 exhibited:

- Ugly URLs, including the “Pages” insert
- Poor XHTML output (thanks to the editor)
- Table-based layout by default — unless you employed a clean, stripped-down master page

In SP 2013, Microsoft has addressed a few of these issues. First and foremost, site managers can now produce search engine-friendly URLs. This feature comes in the form of both automatic URLs based on page and site names, as well as manually manipulated (a.k.a., “vanity”) URLs that allow site managers to change what URL is presented.

Experience	
Publishing	
Standards Adherence	
Multichannel	
Mobile	
Digital Marketing	
Site & Campaign Analytics	
Testing & Optimization	
Segmentation & Personalization	
Social Media Integration	
Promotional Campaigns	
Community & UCG	
Workplace	
Collaboration & Networking	
Dashboard	
Ancillary	
Site Search	
Online Forms	
Module Ecosystem	

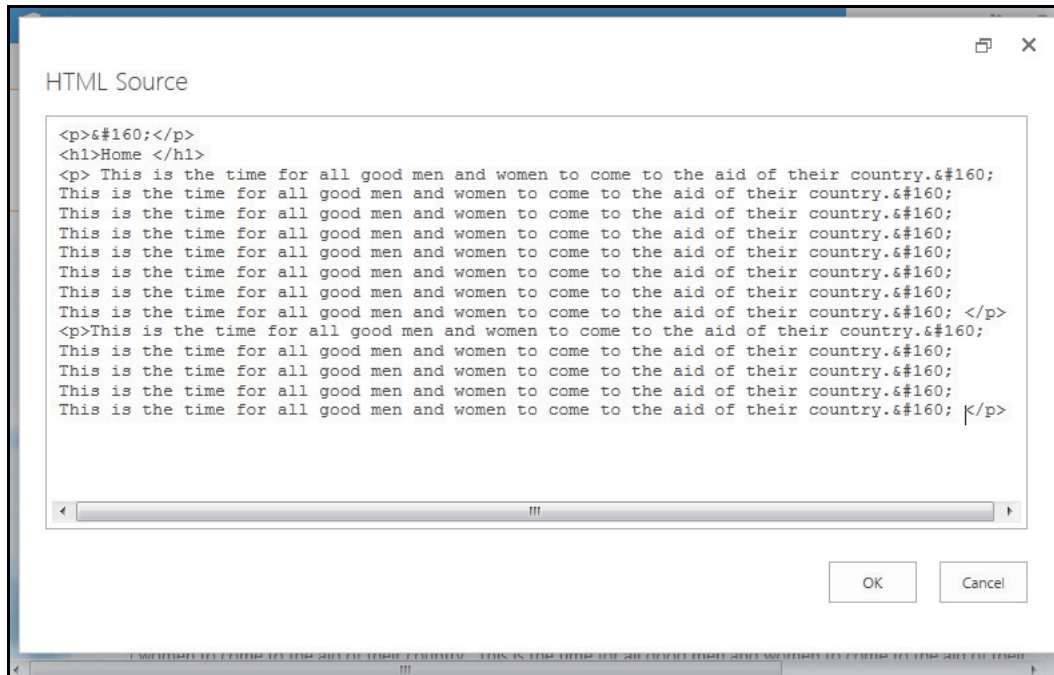


Figure 214. SP 2013 HTML source from the new rich text editor.

In addition to friendly URLs, Microsoft has also added a capability to insert arbitrary metadata, including specific tags to “register” your site with search engines. There’s a new facility that allows site managers to add that metadata easily through Site Settings.

Moreover, the default markup is much cleaner now; tables are gone and it’s now completely CSS driven. This has improved XHTML and WCAG 2.0 AA compliance. Microsoft has created a new dedicated “mobile channel” concept. Mobile-optimized delivery has never been a strength for SharePoint. In SP 2013, Microsoft tried to answer some of the criticism by creating functions and features that enable you to build versions of your site specifically for mobile devices. Specifically, you create distinct master pages for different device channels. Note that page layouts are not device specific, so you’ll need to put some forethought into those and ensure that you are not sending extraneous bits to mobile devices. In particular, you will have to pre-configure Web Part zones carefully if you want Web Parts to appear conditionally, based on device type.

Ultimately, you’re still required to develop a new look and feel for each “channel;” i.e., this is not mobile middleware with toolsets to adjust the experience to a particular device automatically. However, it does offer some useful starter services, including an upgraded version of its mobile device detection module.

Site Contents

Name *
The name used by authors and others to identify this channel

Alias *
Pick a word to identify this channel in code, Device Channel panels, previews and other contexts. Warning: If you later change the channel alias, you will have to manually update Master Page mappings, Device Channel panels, and any custom code or markup.

Description
A quick description of the Device Channel

Device Inclusion Rules *
Specify one or more user agent substrings (for example: Windows Phone OS), placing each substring on its own line. When the user agent string of a visiting device contains any of the specified substrings, the channel will force site pages to display using that channel's optimizations, like a different Master Page or Device Channel Panel. You can also trigger this special rendering by using query strings, cookies or custom code, in which case the substrings don't matter.

Active
Check this box once you've optimized your site for this channel. If you're

Figure 215. Defining a mobile device channel.

Engagement

On the whole, SharePoint remains light here. Remember, this is still a digital workplace and an intranet-oriented platform. Some third-party partners have tried to fill in the gaps here, but they are not well-known companies and the solutions remain very nichey.

With SP 2013, Microsoft has responded with basic improvements over 2010. One of the newest features is the concept of “Follow Content.” This natively allows visitors to subscribe to content within your site. Microsoft has smartly re-purposed the long-time “Alerts” feature, and focused in more of an externally facing way. Unfortunately, field experience with this service remains thin.

Analytics is now handled by FAST. Like its 2010 predecessor however, SharePoint 2013 focuses on basic usage metrics within its site management interface. The metrics include the typical number of “hits” to a specific page.

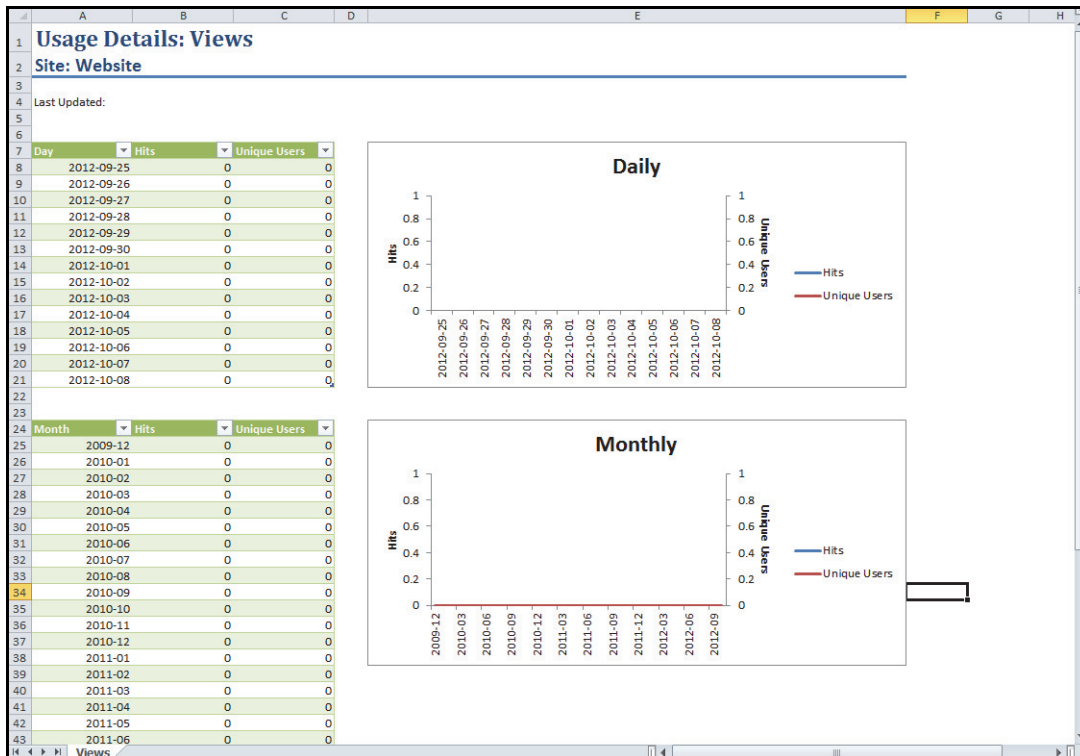


Figure 216. Daily and monthly Usage Details report in Excel.

In general, SharePoint Analytics is not designed to replace what you could get with Google or Adobe Omniture; it's designed for more short-term, real-time data to be fed into dynamic delivery. For example, the new recommendations service runs off Analytics, as well as basic features like "most visited" or "most recommended" Web Parts.

This brings us to a discussion of personalization. Here SP 2013 has definitely improved on SP 2010. SharePoint comes with a recommendations Web Part, which can be set to trigger off various end-user behaviors. You can apply segment metadata to content and use FAST to display certain content to various segments you pre-set. However, the platform here is less wired and you will need to undertake some arcane coding as you did in SP 2010.

Unfortunately, SharePoint 2013 still lacks many of the tools and features marketers desire:

- Advanced previews and testing for personalization services
- Site testing and optimization services
- Bidirectional integration with CRM and marketing automation platforms (including those not owned by Redmond)
- Promotional campaigns, including integration with campaigns run offsite (e.g., dynamically customized Google AdWords landing pages, specialized treatment for visitors via organic search, and so forth)
- Social media site management
- Online forms creation and management

Perhaps most critical among these is SharePoint 2013’s lack of any sort of campaign management functionality. Creating a “campaign” around specific content and tying it to multiple delivery routes (e.g., web, SMS and email) still remains within the realm of add-ons.

A curious lack of integration between SharePoint and other Microsoft business tools is also noteworthy. While Microsoft and SAP have collaborated to create connections between SharePoint and SAP’s ERP solution, you’ll find little in the way of connections between SharePoint and Microsoft’s own Dynamics CRM products. For example, marketers typically want to connect content, campaigns, and other assets to an internal CRM platform. Unfortunately, the Dynamics-SharePoint connection is focused more on internal workflow processes than any marketing capabilities; you can set SharePoint as a storage location for customer-related documents, but you cannot enable a true marketing connection on the public website without custom development. The same is also true for non-Microsoft solutions — think Salesforce — unless you buy a third-party add-on product.

Ancillary Services

For Search, you can use any third-party search engine that will crawl web content, but of course the preferred option is to deploy FAST, which comes with the product and has a rich array of knobs and buttons to tune for both unstructured and structured content.

As a portal platform, SharePoint can boast an extraordinarily wide variety of available micro-applications — typically Web Parts — available either from Redmond or on the open market. However, you see far fewer of these for web publishing.

Vendor Intangibles

As with most products in this report, documentation tends to lag development, and if previous experience is any guide, advice about best practices and workarounds for 2013 are going to take years to roll out.

Another aspect of support is remembering all the details and nuances of SharePoint. In reality, there is an excess of unique details and quirks that ship with SharePoint. You’ll want to keep handy your own FAQ, a running list of bugs, running list of fixes, oddities, and end user instructions. Nevertheless, with 2013, there are so many key switches that need to be thrown at the beginning of an implementation that you need to plan very carefully.

Fortunately, finding good resources is comparatively easy. There is a surfeit of bloggers and forums — within Microsoft and outside of it — all of which are willing to explore the innards of SharePoint with you in a way that you would never find with OpenText/Vignette WEM, for example. The only potential problem here is that most SharePoint specialists tend to be more up-to-speed on the portal and collaboration aspects of the platform, and less conversant in its other capabilities — particularly WCM. Also, the community can be a bit chaotic, and some books and white papers often not only contradict each other, but also eschew Redmond’s best practices. Expect to experiment a bit on your own.

You can download a trial copy of SharePoint 2013 and install and configure in your own environment. Few other commercial WCM vendors will let you do that.

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

There is probably no vendor that has a larger channel to market than Microsoft — and SharePoint has been much hyped, so the number of systems integrators in the field willing to deploy and further develop this product set is larger than perhaps any other vendor in this report.

A challenge for prospective licensees will be finding solid integrators; this is not a simple product. Microsoft has a broad channel, but expertise on a platform this complex understandably deepens slowly. Even if you have strong .NET talent in-house, you should consider aligning with a consultancy that has really gotten inside SharePoint and can help you avoid specific pitfalls. Here, as elsewhere, you'll want to find those (relatively few, we think) specialists who have gotten inside the web publishing pieces to SharePoint, which are quite different from the rest of the tool. Because of the relative immaturity of WCM functions within SharePoint, we see quite a bit of experimentation in the field. SharePoint can boast some public websites under its management, but it has far fewer than most of its major competitors (like Ektron, EPiServer, Kentico, Sitecore, Telerik, DotNetNuke, et al.).

This brings us to the bigger question about the product's three-year development cycle. Web experience management will be significantly different in 2016, but will SharePoint be able to adapt? Redmond says it is pouring most of its innovation into the SharePoint cloud offering, but Office 365 remains almost prohibitively crippled for a public-site web publishing scenario.

SharePoint pricing is a bit complicated, but to its credit, Microsoft is very open about it. It comes in three editions: Foundation, Standard, and Enterprise. Web content management comes automatically with the Standard and Enterprise editions. However, for more advanced features like Cross-Site Publishing, you'll need the Enterprise Edition.

SharePoint sells via a combination of server and client access licenses ("CALs"), listing at about \$7,000/server and about \$100/seat per user or device. Note that you no longer need to purchase an external site license, which previously made SharePoint almost prohibitively expensive for public sites using SP 2010. Thus, you will likely need more and better servers for 2013. This is a hardware-intensive platform, and Redmond increasingly encourages you to separate different services onto different servers. Moreover (as before), you cannot share a public-facing farm with an internal-facing farm, so there are no savings there. You must apply full licenses to any failover or staging environments (but not dev: that's handled by MSDN licensing). While you'll likely find SP 2013 less expensive than before, but it's still not cheap or slim, like most of its .NET competitors.

Conclusion

In general, the product still feels like a web publishing tool shoehorned uncomfortably into a document management system — but redeemed in part by its potentially very powerful development facilities. A talented integration team or Microsoft partner with deep .NET skills can learn to build sophisticated applications with SharePoint 2013. The only problem for you is to ensure that the development, testing, and propagation of additional Web CMS-specific widgets — through the broader Microsoft ecosystem — will be measured in years and not months. There is a serious disconnect here with Redmond's marketing, which pushes the product as a very out-of-the-box solution.

The good news is that, for web publishing, SharePoint 2013 is an improvement over previous versions — but an improvement more in degree than in kind. Microsoft has addressed some long-standing customer complaints by adding features like image renditions, better mobile

detection / mobile site development, and an improved editing experience. At a time when “digital marketing” is driving investment in public sites, Redmond seems to have a tin ear for the concept.

To that extent, the real test comes when comparing SharePoint against its peers. Compared to more agile competitors, SharePoint 2013 still shows Microsoft’s lack of focus on WCXM — particularly for public-facing sites. The fact remains that SharePoint still lacks significant marketing and experience management capabilities. More generally, SharePoint 2013 out of the box is still geared more toward the developer or integrator than it is toward the marketer. Thus, our long-standing advice about SharePoint for WCXM will remain the same; for public-facing websites, you should consider other alternatives in a truly competitive selection process. For the internal digital workplace, SharePoint 2013 could offer a good alternative for managing unstructured HTML content, but primarily within the context of other internal collaborative efforts you’re already undertaking with the SharePoint platform.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Plone: Plone

www.plone.org

Project at a Glance

Specsheet	Plone: Plone 4.1 Summary
Geography	Global
What's New	<ul style="list-style-type: none"> • New commenting system
Strengths	<ul style="list-style-type: none"> • Hybrid CMS + portal structure works well for website-in-a-box use cases • Strong support for XHTML and CSS, with strong accessibility bent for content contributors and consumers alike • Multiple authentication options • Resource compression for fast page transfers and bandwidth conservation • Natively generates friendly URLs • Highly tunable, policy-driven workflows • Support for clustering and “high-availability” architectures • Unusually broad set of multilingual content management tools • Large number of third-party add-on modules available • Unusually broad developer community with very mature governance structure, active forums, and a surfeit of online resources • Permissive GPL v2.0 license, with no plans to move to GPL 3.0
Weaknesses	<ul style="list-style-type: none"> • Many moving parts, resulting in a steep learning curve for administrators, power users, and developers alike • No native support for any database other than “ZODB” • Customization requires significant Python expertise • Has no native API, and third-party modules vary hugely in quality • Coupled development, management, and delivery architecture can be problematic for managing large and/or busy sites • Version upgrades for Python or Zope (foundational technologies for Plone) can have unpredictable ripple effects throughout the CMS • Developer intercession is required to create and modify templates • Project momentum feels slower than competitors in past year
Potential Fit	Informational Site, Microsites & Landing Pages, Community-Oriented Site
Unlikely Fit	Advanced Marketing Portal, Multichannel Publishing; Ultra-Large Single Site
Compare To	Drupal, eZ Systems, TYPO3
OS	Windows, Linux, UNIX, Mac
Repository	Zope object store database (ZODB)
Client	Browser, WebDAV, FTP, External editor
App Platform	“Zope” application server
Licensing	GPL 2.0
Ownership	N/A; “Plone Foundation” serves as project leadership directorate

Summary

Plone is the largest in its class of Content Management Systems written on a Python-based framework called “Zope.” Plone is among the oldest, most widely known, and best supported open source WCM systems. Managed by the Plone Foundation, which was established in May 2004, the CMS itself has been under development since 1999. The Foundation — which owns the copyrights and trademarks associated with Plone — sets the overall direction of the project, providing the strong leadership that’s lacking in so many open source projects (notably Joomla!).

In terms of support for things like globalization and localization, accessibility, and policy-driven workflow, Plone is comparatively sophisticated (as open-source WCM systems go), and thus

Plone is frequently used as a foundation for accessible, customized, and quite sizable websites, ranging from corporate brochure sites to ultra-large single sites and e-commerce websites. (Adopters include such organizations as NASA, the CIA, and the Free Software Foundation.) Given the right amount of Python development talent, Plone is flexible enough to enable quite a diverse range of web-based applications. Natively, however, Plone wants to serve as a collaborative workspace, knowledge base, or basic informational website.

You’ll almost certainly need at least one Python-savvy developer just to get your templates in order (even on a relatively uncomplicated site). If you want Plone to behave like a higher-end Web CMS — with the ability to scale up and out, take advantage of load balancing, and so on — you’ll want to budget for a platoon of Python ninjas. Plone is as much a platform as it is a set of WCM applications; what you make of it is up to you and your developers.

Introduction

Plone is built off the Zope Content Management Framework (CMF), which provides core content services such as workflow, content indexing, and content types. While CMF is a powerful framework, it is not an out-of-the-box CMS solution that can be rapidly installed and customized. For this reason, three developers — Alan Runyan, Alexandar Limi, and Vidar Andersen — started the Plone project in 2001. A vibrant community quickly formed around the project, and today the ecosystem includes thousands of users and code contributors, third-party commercial support, training classes, and an extraordinary amount of documentation (including half a dozen trade paperbacks that you sometimes can still find at your local bookstore). This easily makes Plone the largest Zope-based Content Management System — and possibly the best-known truly free open source CMS, period.

Plone is a comparatively well managed open source project, especially considering the size and scope of the developer community. (Contrast this with the Joomla! project, which is more

Scenario Fits	
Simpler Site	
Informational	<input checked="" type="checkbox"/>
Microsites & Landing	<input checked="" type="checkbox"/>
Mid-Range	
Basic Digital Marketing	<input type="checkbox"/>
Mobile Site	<input checked="" type="checkbox"/>
Community Oriented	<input checked="" type="checkbox"/>
Complex Site	
Advanced Marketing Portal	<input type="checkbox"/>
Global Enterprise	<input type="checkbox"/>
Multichannel Publishing	<input type="checkbox"/>
Ultra-Large Single	<input checked="" type="checkbox"/>
Digital Workplace	
SMB/Departmental	<input checked="" type="checkbox"/>
Global/Enterprise	<input checked="" type="checkbox"/>

like constrained anarchy.) Pending enhancements, called PLIPS (Plone Improvement Proposals), are prioritized and grouped into releases — done in “sprints” — that have a designated primary and secondary manager. There are also teams (sub-projects) that manage documentation, internationalization, user interfaces, and performance optimization, among other things. Runyan and Limi still play an active role in the community, and several other leaders have emerged.

Plone’s underlying technology platform, Zope, has gone through a considerable amount of change over the past years. Version 4.1 was released in August 2011, and is based on Python 2.6 and Zope 2.13. On the one hand, Plone has more dependencies on the underlying Zope and Python stacks; on the other hand, Plone can benefit from the improvements to those technologies.

Technology

Technical Administration and Security

Plone is built on top of Zope, the Python-based application server. Zope, which stands for “Z Object Publishing Environment,” is to the Python programming language what a J2EE application server is to Java. It provides a managed runtime container for Python code, plus infrastructural functionality, such as database connectivity, protocol support (HTTP, WebDAV, and FTP), user session management, and administrative facilities. In addition, Zope provides built-in user and group management, with granular access control, and other desirable functionalities.

For better or worse, Zope’s persistence layer (and thus Plone’s, as well) is based on a uniquely Pythonesque object database called ZODB (Zope Object Database), which — depending on your point of view — is either a tremendously powerful way to manage object-based content, or an annoyingly arcane departure from standard RDBMS technology (although you can still use an RDBMS, which we describe later). The good news is, ZODB is transactional, reasonably scalable, well proven in practice, and can sometimes outperform SQL-based query engines.

ZODB itself can be stored in variety of back-ends including file systems and a relational database. Think of it like a Java Content Repository (JCR), where in the Java world, the repository can reside on a file system or database. According to Plone, the reason this feature exists is because (in some scenarios) organizational policies and business reasons dictate how you store your data. For example, some organizations mandate that all data is stored in an RDBMS such as Oracle. If you do choose to use an RDBMS for storing ZODB, you still can’t use ZODB in “relational” terms, although you can use your database’s tools for backup, restoring, replication, clustering and other system administration features. In any case, you

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

can't use SQL with ZODB; instead, you must access it programmatically with Python. Thus, your SQL skills are not reusable.

Bottom line: ZODB is strictly a one-off technology, not used anywhere outside the Zope world. More generally, the somewhat ungainly Zope-based architecture has a vaguely proprietary feel to it, and its lack of adherence to familiar industry standards can make it difficult to integrate non-Python pieces with the system.

With Python, you (or someone in your organization) will learn Python if you intend to make effective use of Plone. On the plus side, Python is a very powerful — and quite popular — scripting language, justly renowned for its rapid application development prowess. Although it lacks the élan of Ruby, it's a well-proven language with an astonishingly wide following. It won't be going away any time soon. Unfortunately, experienced Python programmers that have written large, complex applications in the language are not always easy to find.

A significant improvement that came into play with version 3.2 is the use of so-called “buildout” architecture. In Plone parlance, a buildout is a type of self-contained environment where you can manage dependencies (involving Zope, Plone artifacts, and/or third-party modules) separately from the Zope instance itself. This offers numerous advantages, including making it possible to reuse configuration resources and “eggs” (Python modules) across sites and easier integration of non-Python resources.

Another significant architectural enhancement to Zope (hence to Plone) was introduced in version 4, which is the ability to store large files in the file system, rather than as BLOBs in the database (ZODB). Note, however, that even though the files are stored on the file system, they are still managed via ZODB. As a result, the database size is much smaller because the BLOB is not part of the main database file, but you still get the advantages of support for transactions, clustering, and ACID capabilities of ZODB. There are also significant implications on performance, as discussed later.

Basic installation is gratifyingly straightforward: operating system-specific installers are available for all popular platforms. Once installed, a new Plone site can be created from the administration interface, and the result is a website that looks almost exactly like plone.org.

Nevertheless, setting up a production deployment still requires considerable expertise, time, and planning. In addition to setting up caching (see discussion under Performance further below), you'll want to consider a data-replication and backup strategy, since so much depends on the health of the single data file that Zope uses to store its own content and configuration. (ZODB replication is straightforward, however.)

The system is entirely object-oriented. Code, content, and everything else within Zope is an object in Zope's folder structure, each with its own URL from which methods are invoked. The Zope “acquisition” model causes sub-trees to inherit objects and properties from their parents, which is a familiar enough concept in the context of, say, access rights trickling down through a directory hierarchy, but the Zope implementation can feel more than a little foreign. For example, if a folder has an object in it (such as an image), that object will be accessible in sub-folders, no matter how many levels deep. This means that the paths /folder/image.jpg and

folder/sub_folder/image.jpg will point to the same file, unless /sub_folder happens to contain its own image.jpg file.

The counterintuitive “Acquisition Algebra” upon which Zope’s contextual storage system is based is considered esoteric even in academic circles, but it rests on solid theoretical foundations and offers a number of advantages. (Thankfully, it’s much easier to use than it is to explain.) Nevertheless, this is the kind of thing that — once again — marks Zope and Plone as more than a little quirky.

Unless otherwise configured, all objects are automatically stored in the ZODB (except for large files, which are stored on the file system but are still managed via ZODB). This serves to reduce the amount of code that must be written when developing on Zope. However, using ZODB for storage (or managing storage) essentially eliminates the ability to integrate with other technologies such as Java, PHP, and .NET. Zope does support SQLAlchemy, a Python toolkit for accessing SQL databases.

To be clear, merely putting SQLAlchemy in front of Oracle doesn’t mean Plone or Zope is going to use Oracle for native system operations. If you need to modify core system behavior, you’re back to dealing with ZODB code natively.

The fact that Plone is based on a one-off stack may cause some issues if you plan to use Plone in conjunction with other products in your enterprise. In fairness, the Plone community has concentrated on making the platform easier to integrate with other applications. Plone 3.0 incorporated an “events” service (thanks to the new events framework built into Zope), so that calls to other systems can be triggered when content is moved or saved. While some users have integrated Plone with salesforce.com, SugarCRM, Facebook, and Twitter, you will mostly need to develop any integration on your own.

Plone itself does not offer APIs to expose its content and services. A third-party project called `wsapi4plone` is an attempt to mitigate this by providing a Web Services interface over Plone. This provides an XML-RPC (or REST-based) API to access Plone. However, the usual disclaimers about third-party modules are applicable here.

With respect to publishing, Plone is by nature a tightly coupled management and delivery system. As with Joomla! and Drupal, you rely on the same application to manage content as well as to serve it up. This means that, out of the box, Plone does not support the notion of a staging server or of deploying content to a production server. Everything is done on the production environment, which may be a cluster of identical instances, all talking to the ZODB. However, that doesn’t mean there aren’t workarounds for situations where multiple tiers are a necessity. Sites with major security concerns typically go to some lengths to run Plone within a firewall and use technologies like CMFDeployment to push static HTML files to a public-facing web server. High-traffic websites like Novell’s will typically use a hybrid model, where some pages (e.g., the home page) are statically deployed. Other, more dynamic and less-trafficked pages are served directly from Plone (through caching technologies like Squid or Apache `mod_cache`).

For deployment that is more sophisticated and decoupling, some customers (like the CIA) use Enfold’s commercial “Entransit” product, which enables you to publish structured content out of a Plone instance to another delivery tier built on a different technology.

Plone has a flexible authentication system (part of Zope) called Pluggable Authentication Service (PAS). PAS enables you to break your authentication process into smaller steps. This is useful for organizations that require authentication against multiple systems. Instead of replicating data from multiple systems, you actually leave the data where it is and use it at runtime. You also can choose to cache it. Plone offers a variety of plugins to authenticate against different sources, such as LDAP, Active Directory, OpenID and more. However, remember to test it against your specific scenarios. It is possible that the plugin for your specific source is not available, or is inadequate, in which case, you will have to write your own plugin. Nevertheless, the framework is available and is quite flexible.

For access control purposes, the system supports users, groups, and roles. Groups are named collections of users. Each user or group can be assigned one or more roles, which (in turn) are mapped to permissions.

In terms of the user interface for managing users, roles, groups, and permissions, you don’t have to use the imposing UI of Zope Management Interface (ZMI); instead, you can use the user-friendly Plone site admin UI.

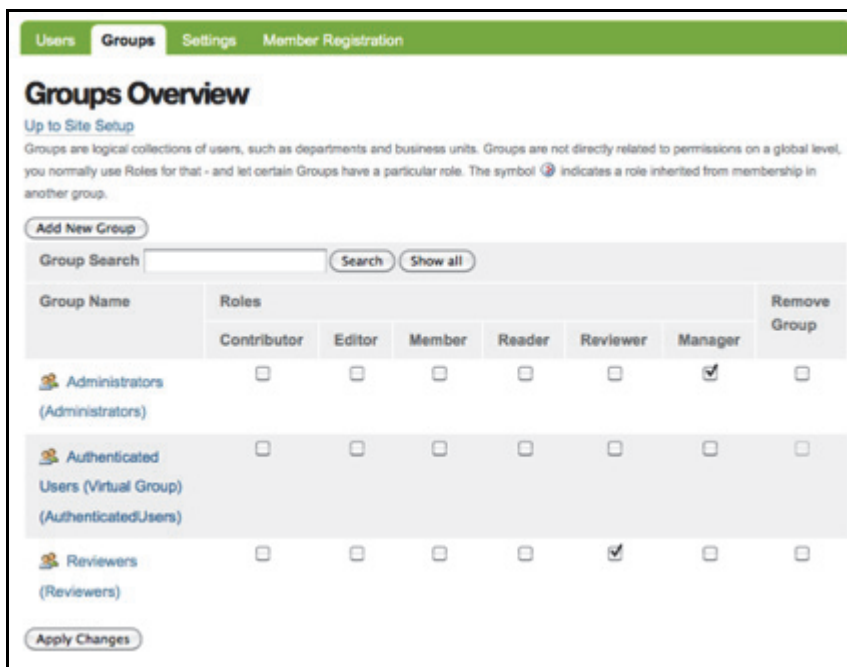


Figure 218. You can now manage Users and Groups from within Plone.

Interestingly, the system supports a resource-sharing capability, wherein an author or editor can make content in one of his or her private folders available to another individual or group. While this functionality has been part of the product for a long time, many day-to-day users have had a difficult time understanding or making proper use of it. Plone 3 updated the Sharing tab to make the functionality more visible to the casual user, and you now can go to a

specific folder in your site, and give sharing permissions directly from there. Again, this may or may not be a good thing; you may find that it requires training (at the very least), and could it complicate governance, as well.

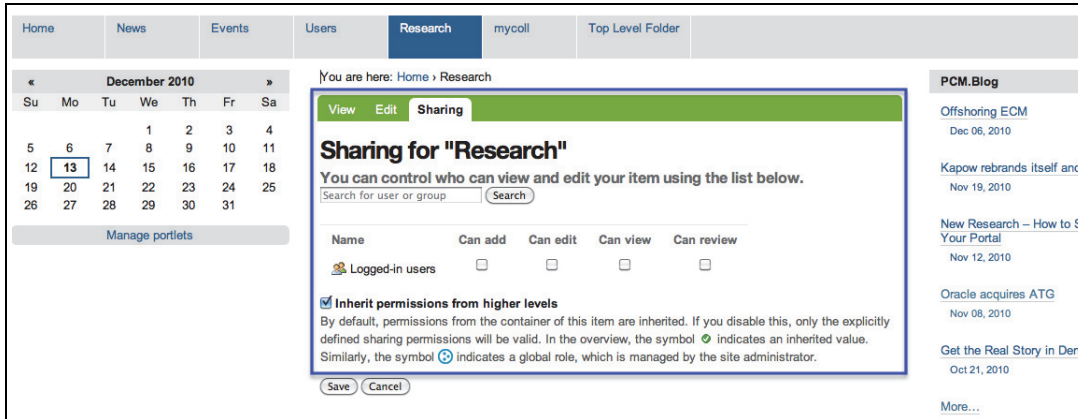


Figure 219. You can select who can do what directly from the Sharing tab under each folder or content item.

The default roles (which you can augment with your own custom roles) include Anonymous, Authenticated, Manager, Site Admin, Member, Owner, and Reviewer. Of these, Site Admin was introduced in the recent 4.1 release. The Site Admin role allows users to perform site administration activities like adding and updating site users and groups. In previous versions, you would have to grant such users manager’s access, which would also allow them to restart the server or change system configurations.

Note that since system permissions can be assigned only to roles (not groups or users), there’s a *de facto* requirement that everyone be in at least one role. Note also that individuals can be assigned more than one role. There’s no need to sign in and out again, in order to switch roles, but similarly, the ability to be in more than one role at once is generally frowned upon in highly governance-oriented environments. You may have to do some careful planning and/or system alteration to support your governance requirements; take note.

The other thing to understand about roles is that Plone supports a model of *global* versus *local* roles. Global roles are meant to correspond to site-wide, high-level permissions, involving things like visibility constraints on large classes of objects for a given role. By contrast, *local* roles are contextual: They are granular to the level of folders, content objects, and even individual components within a page. A knowledgeable implementer can set permissions on workflow payloads to control the role-appropriate visibility of content items at different points in the workflow. The system is flexible, and the flexibility derives directly from the peculiar “acquisitions-based” resource-addressing scheme. However, flexibility is not always a good thing when it comes to security models. In general, systems with this kind of fine-grained, inheritance-based role and rights management tend to become implementation and administration nightmares. Our advice would be to spec out your rights model carefully, in detail, ahead of time; draw upon only as much of the system’s power as you need. Otherwise, you risk creating an entitlements quagmire.

There is no native reporting in Plone, outside of the raw weblogs. User information can be tracked by adding new properties to the `_memberdata` table. For example, this could be used to

track the number of logins. Like other CMS tools, you can query the Zope database for information using Python scripts, which then create text or CSV files, which you could integrate with existing reporting tools. Beyond that, there are no native reporting tools, and there are no reporting modules in the pipeline at the time of this report. The Plone team recommends using external systems for reporting.

Development

Plone can be customized in a number of ways. It's generally considered a best practice to do all development as "products" on the file system, which enables code to be managed by a source-code management system and moved between servers easily. Typically, an implementation will involve one "product," which is used to store all the customization to the Plone skin, and then another product for each distinct piece of functionality. This strategy enables functionality to be collaboratively developed by a team working on local development environments and reused across other Plone sites. More ad hoc or experimental development can be done through the web interface by overriding default behavior and layouts in the "custom" folder or adding page templates through the ZMI. However, code written in this way is stored only in the ZODB, where it is less reusable and more difficult to manage.

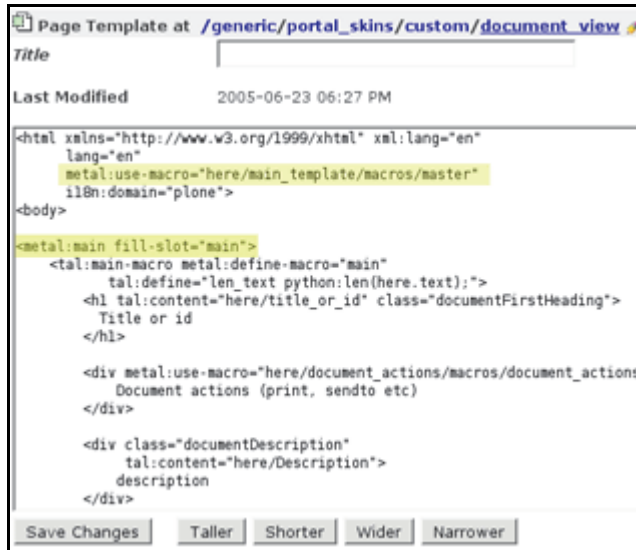
Individual folders and entire Plone sites can be exported and imported between Zope instances, making it relatively easy to move content from one Zope instance to another. However, Zope's undo functionality, which allows updates to be made to a production site, while remaining invisible until deployment, is often considered cumbersome and is rarely used.

Plone's preferred operating system (employed in most Plone implementations) is Linux. Certainly, this is the platform with the most readily available support, and it enables the use of commodity hardware. However, Plone also runs on Windows, UNIX (Solaris, AIX, and others), and Mac.

In terms of content modeling, Plone recognizes the following native content types: *page*, *event*, *file*, *folder*, *image*, *link*, *news item*, and *collection* (which is like a dynamic saved search). All content types have extended metadata properties, including most Dublin Core elements.

Plone's content modeling framework is based on a framework called Archetypes. The framework automatically generates an edit form, view template, security, and storage for the new content type. While it is a powerful framework, it is also resource hungry and requires considerable developer bandwidth. You need to work with Python code and classes to model content types. Much of this work is automated through the third-party ArchGenXML product, which can take a UML diagram as input (authored in the diagramming tool, Poseidon), and auto-generate all required Python code and scaffolding so Plone will recognize it.

Archetype objects are considerably heavy and loading them from the database affects performance. Plone is working on another framework called Dexterity (it has been introduced in version 4) that is more efficient and lightweight. You can use it now, but it has not been internationalized yet. Although better than Archetypes, you will still need technical people to create new content types. There's still a long way to go before there's a simpler, web-based interface to create new content types without writing new code.



```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
lang="en"
metal:use-macro="here/main_template/macros/master"
xmlns:domain="plone">
<body>
<metal:main fill-slot="main">
<tal:main-macro metal:define-macro="main"
tal:define="len_text python:len(here.text):">
<h1 tal:content="here/title_or_id" class="documentFirstHeading">
Title or id
</h1>
<div metal:use-macro="here/document_actions/macros/document_actions"
Document actions (print, sendto etc)
</div>
<div class="documentDescription"
tal:content="here/Description">
description
</div>
Save Changes | Taller | Shorter | Wider | Narrower
```

Figure 220. Page templates use a macro framework.

Page Templates use a macro framework called METAL (Macro Expansion Tag Attribute Language) that enables reuse of rendering logic between templates. A macro can define a region called a “slot” that the calling template can fill with its own text. This mechanism makes macros highly reusable. Pages on a Plone site are rendered by calling the content type’s default presentation template which, in turn, calls the Master macro within the main template with instructions to fill the content area slot (typically in the middle of the page) with the presentation logic from the calling template.

Advanced presentation and business logic is implemented within logic objects using Python code and page templates. Logic objects execute logic (such as form processing, string manipulation, and calculations) or access external resources and return results that presentation templates display.

There are three strategies in customizing the look of Plone: adroit use of CSS, changing the existing page templates, or replace the rendering framework with your own. The CSS method is the most common and often the most practical. Plone uses XHTML markup that relies heavily on style sheets for colors, typeface, imagery, and element positioning. For deeper customization, the existing page templates can be overridden by a style product. The Main Template (which orchestrates all the components of the page) is complex because it supports all of Plone’s collaborative portal features. The temptation is to remove all of the unneeded code; don’t do it. In addition to making an upgrade difficult, a lot of work was done to make the generated pages both WCAG-compliant and search-engine friendly. The third strategy, of replacing the rendering framework, creates two sites: a Plone-looking site used for managing content, and an externally facing site, which uses a very different set of templates.

If you decide to create your own templates, there are two common approaches. The first approach is uses what is known as Diazo (it was called Collective XDV previously). Diazo enables you to use static HTML and CSS as a theme file and use rules that you define in an XML file to dynamically populate parts of the file. It is essentially a way of transforming Plone’s output to your chosen look and feel, using the rules as defined by an XML file. In this way, you can theme a site without actually using any Plone-specific code.

The second approach uses an external product called ZopeSkel with the Paster templating system. ZopeSkel automatically generates a basic Python package for your theme. This consists of a directory with all of the required files and hooks you need to modify or create your own theme.

Most developers use tools like Dreamweaver or UltraEdit to work with templates. As is the case with most open source tools, there are no tools to make template modification user-friendly for non-techies.

Performance

Performance is always a tricky area, and this has been a concern for Plone as well. For example, while the ZODB performs well with average size repositories and under read-intensive loads, Zope implementations with large repositories (greater than 10 GB), or that are write intensive have had performance issues in the past. Other aspects that affect performance are related to certain Zope dependencies, its tightly coupled architecture, and Python itself, which has a somewhat primitive threading model. Whereas Java's JVM can efficiently manage many concurrent threads and take advantage of powerful multi-processors, the Python Virtual Machine has no notion of thread priorities, nor can threads be destroyed, stopped, suspended, resumed, or interrupted. This is a serious limitation.

Various caching and clustering strategies, as well as "baking" major portions of a website (the sort of strategies you'd employ in any tightly coupled system, in other words) have been effective in mitigating this performance limitation, but Zope doesn't natively lend itself to high-transaction, volume applications, which is why you should invest time to understand its caching options.

Although Zope has its own HTTP server, any public-facing Zope website serving anything but trivial amounts of traffic should sit behind a caching proxy server for performance and security reasons. Plone now ships with `plone.app.caching`, an HTTP response caching system. It replaces `CacheFu` and according to Plone, it is more configurable. However, it is quite new; test it well.

Plone has made significant improvements in version 4. Benchmarks published on Plone.org show that Plone 4 is faster than its predecessors are, but you should validate these benchmarks on your own. Plone's version 4.1 is now based on Zope 2.13, which features further performance improvements specifically to ZODB and ZCatalog (Zope's built-in search engine).

With deft use of caching proxies, Plone can be taken to surprising heights (as proven by Akamai and others), but it's always a nagging concern. We urge you to test this aspect well.

Content

Contributor Experience

Usability is a mixed bag. While Plone’s basic contributor interface is relatively simple and easy to use, many complain of a steep learning curve for administering, configuring, and integrating a Plone site. This is only partly a UI issue; it is mainly due to a large stack of new technology under the hood that you occasionally need to dip into in order to accomplish certain tasks. For example, much of the configuration is done through the Zope Management Interface (ZMI), which new users often find difficult to navigate because of the veritable zoo of folders in which components are tucked away. Plone has mitigated some of these issues by moving many administration tasks into the Plone (rather than Zope) layer, but the ZMI still lurks underneath.

Plone was the first Content Management System in the world to be compliant with the WAI-AA and US Section 508 accessibility standards, and is still a market leader in this regard. The CMS itself is 508 compliant and web pages produced in the system are, by default, accessible (aided by the system’s traditionally very strong support of XHTML 1.0 and CSS). However, accessibility is largely what you make it, too; even a system as strong for accessibility as Plone can be defeated through poor page design.

Multiple language support is elegantly handled through Plone’s internationalization framework and an active translation community, which has contributed over 50 localizations of the product. (The original Plone, version 1.0, came in 22 languages.) In general, the product itself is quite strong in areas of internationalization and localizability.

Contributing Content

Like Drupal, Plone has an in-context editing model that merges many management functions into the externally facing website. If you are an authenticated user navigating the site, you’ll see a set of tabs that expose interfaces to edit site content. You can also use an in-line editing model where you do not have to click the Edit tab to make an edit. Each content attribute turns into an input field when an authenticated or an authorized user hovers over it. Assets are also updated on a field-by-field level rather than at the whole asset level.

Note that inline editing is disabled by default. Therefore, many new Plone users were inadvertently activating edit zones (and becoming confused by it), it was decided that the feature should no longer be enabled out of the box.

Plone offers several options when authoring a document.

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

image references. Note, however, that there is no detailed impact analysis-reporting feature; all you get is what's shown in the pop-up warning. This tends to mitigate against broad, object-oriented content reuse. Similarly, you will not find any native retention or records management facilities.

Content Lifecycle

Workflow in Plone is based on Zope's DCWorkflow product and uses an extensible model of states and transitions. States determine the visibility of assets and what actions can be taken on those assets. Plone comes with various workflows, allowing content to be kept private, reviewed, and published in various ways. It is also relatively easy to create your own workflow and security policies.

Transitions control the shift of assets between states and provide events to trigger scripts (such as emailing and flushing cache). A person in an approver role gets to see a portlet that lists pending approvals. In earlier versions of Plone, configuring workflows required working in the notoriously complicated Zope Management Interface and writing a fair amount of Python code. Plone 3 provides an easier-to-use interface to manage workflows, and it provides more out-of-the-box, multi-step workflows.

It has become easier to maintain sites with content in multiple languages using Plone, because as Plone allows the language-specific sites to be decoupled, the translated content can be coupled, if required. For more sophisticated translation requirements, a module exists that makes it possible to drag and drop an entire site — or any part of a site — back and forth via the XLIFF-format.

The default Plone site is immediately usable if you want a collaborative workspace à la Microsoft SharePoint or IBM Quickr, or a basic informational website. It is frequently used as a workgroup collaboration portal. Registered users can upload articles, files, calendar events, and web pages into personal and group folders that are searchable using Zope's indexing and searching functionality. An AJAX API for search is included, and that makes it easier for editors and content contributors to find a content object from the repository. AJAX can also be used to re-order items in a folder.

There are no built-in retention services in Plone.

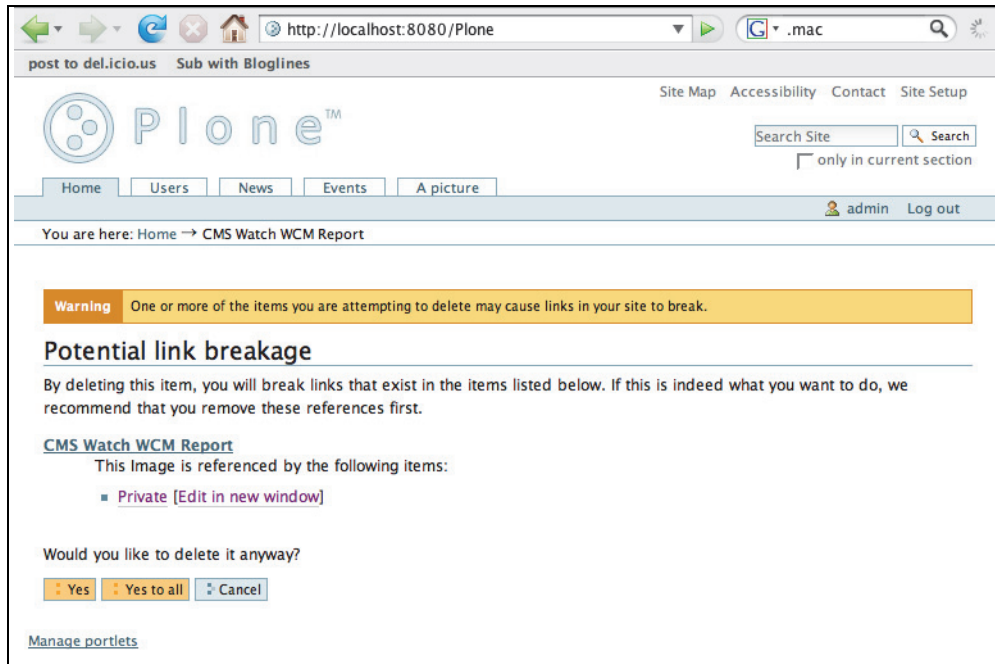


Figure 222. Plone 3 adds dependency management to links and embedded images.

Experience

Publishing

Like eZ Publish, by default Plone creates user-friendly URLs such as:

<http://plone.org/documentation>

Digital Marketing

There are few sexy experience management and digital marketing services here. For web analytics, you can paste the tracking JavaScript in your template and there's an out-of-the-box integration available with Google Analytics.

Plone's social media and collaboration capabilities remain quite elementary. There's a commenting framework that provides features like moderation, email notification, CAPTCHA support, and spam protection. As always, if you have integrated a third-party commenting package and would like to migrate to the native commenting system, you might need to plan for manual migration of your data.

Experience	
Publishing	
Standards Adherence	<input checked="" type="checkbox"/>
Multichannel	<input type="checkbox"/>
Mobile	<input type="checkbox"/>
Digital Marketing	
Site & Campaign Analytics	<input type="checkbox"/>
Testing & Optimization	<input type="checkbox"/>
Segmentation & Personalization	<input type="checkbox"/>
Social Media Integration	<input type="checkbox"/>
Promotional Campaigns	<input type="checkbox"/>
Community & UGC	<input type="checkbox"/>
Workplace	
Collaboration & Networking	<input type="checkbox"/>
Dashboard	<input type="checkbox"/>
Ancillary	
Site Search	<input type="checkbox"/>
Online Forms	<input type="checkbox"/>
Module Ecosystem	<input type="checkbox"/>

Ancillary Services

Although Plone comes with its own Search facility (which has a type-ahead and live preview capability that puts preliminary results in a small widget overlay as you type), there’s no shortage of third-party search add-ons for Plone, some of which — like PloneSearchBox — are portlets. The advanced search page now uses jQuery to hide or un-hide different options dynamically, based on your permissions or relevance. It also now searches East Asian languages better.

Plone’s “portlet” framework makes it comparatively easy to add micro-applications (such as a calendar or a task list) through Plone’s user interface. Many add-on modules are implemented as portlets because of the ease in which they can be integrated into the UI.

Plone 3 and higher makes it easier to manage which portlets appear on the page and where. Thus with respect to micro-applications, the story with Plone (as with other open source packages), is “find a module.” There are scores of add-on modules for forums, blogs, wikis, event calendars, and so forth.

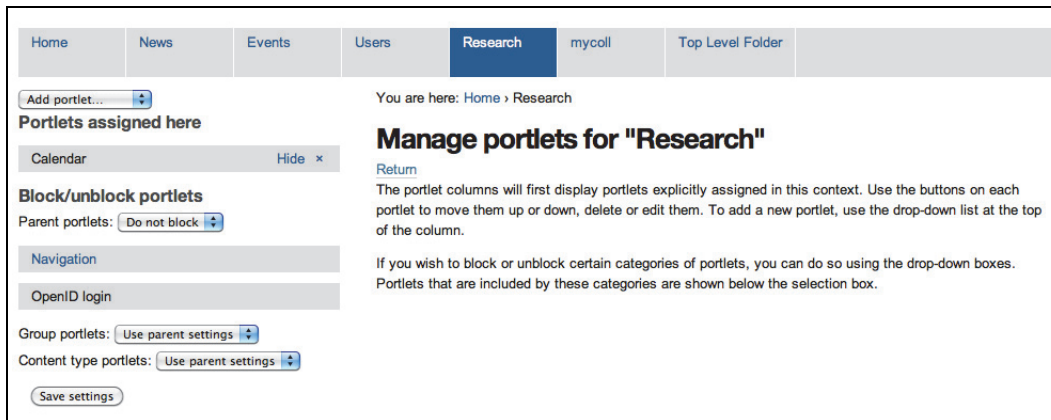


Figure 223. Managing portlet positioning in Plone 3.

Project Intangibles

Around the world, a sizable network of mainly small boutique consulting companies has sprung up to provide maintenance and support for Plone installations. Many of these firms also promote various modules that they have developed and contributed to the community, or are selling to customers. As a practical matter, Plone integrators tend to be smaller and more narrowly focused than consultancies that specialize in other open source CMS tools. Unfortunately, Plone has no partner certification program, which means it can be hard to find truly qualified help.

Owned by the Plone Foundation, which was established by a group of major Plone integrators and customers in May 2004, Plone itself has been under development since 1999. The Foundation (aided by commercial sponsors like Computer Associates) has done a lot to

Intangibles	
Vendor Services	
Vendor Professional Services	<input type="radio"/>
Channel Partner Services	<input checked="" type="radio"/>
Support & Community	<input checked="" type="radio"/>
Strategy & Roadmap	<input type="radio"/>
Viability & Stability	<input checked="" type="radio"/>

promote and protect Plone over the years. In a governance sense, Plone is undeniably one of the better-managed open source projects in the CMS space.

As with most open source projects, Plone's documentation has been criticized for its lack of quality and depth. This is less true today than, say, a few years ago, but it's still an issue. One problem is that when a system contains as many complex (and in most cases, non-standard) foundational technologies as Plone does, no one piece of community-written documentation will cover everything that should be covered. Yet, the architecture's many dependencies are such that you almost *have* to have documentation that crosscuts the various underlying pieces.

On the other hand, there are a number of hard-copy books on Plone, which are especially helpful when used in conjunction with the Python programming language tutorial on the Python.org site. What the documentation does not cover can usually be found on the Plone mailing lists (of which there are a dozen), archived and searchable on GMANE.org.

Both the *plone.org* and *zope.org* websites host libraries of components that anyone can search and download (more than a thousand of them currently exist). These extensions vary significantly in maturity, but in general, the authors are candid about their capabilities and stability. Most extensions are open source, but there are also some non-open source products such as Enfold Systems' Plone Desktop and Enfold Enterprise Server — which bundle Plone with professional support and several commercial add-ons such as IIS integration.

The key question is whether the Plone community can maintain interest in Zope. In absolute terms, the number of commits per day to Plone's development repository has been on the rise. However, there's competition from technologies like Ruby on Rails, Python application frameworks (such as Django and TurboGears), and PHP frameworks (such as symfony, Zend, and eZ Components).

When you look at metrics like blog mentions, Alexa site rankings, and social-bookmarking counts, Plone consistently trails far behind Joomla!, Drupal, and WordPress. In addition, Google search-query counts for Zope and Plone have steadily dropped (by something like 75 percent) in the past years. While we certainly don't think Plone is in danger of going away any time soon, it's nevertheless possible that Plone mindshare (if not adoption rate) has peaked — and may even be turning downward. Perhaps it is reaching the natural limits of global Python adherents.

One of the benefits of a well-managed, open-source project is that the roadmap is posted (and discussed) online, so if any major changes are planned, you have some advanced warning. The roadmap is publicly available at:

<http://plone.org/products/plone/roadmap>

We advise you to watch the roadmap page (and the [Real Story Group blog](#)) regularly, as Plone's roadmap (and progress along it) tends to be a moving target, seldom looking the same from one month to the next.

Conclusion

Although there are many external and internal sites that use Plone straight out of the (virtual) box, Plone is more often used as a foundation — a platform — for highly customized websites, ranging from corporate brochure sites to e-commerce sites. Using Plone as a starting place also opens the door to many Plone-based third-party products in addition to the huge number of available Zope-based products and tools. However, as with Drupal and Joomla!, the platform’s “sweet spot” still revolves around community-oriented portals.

With minimal configuration, Plone can serve as a highly capable collaborative workspace, knowledge base, or basic informational website. In the hands of a skilled Python developer, Plone can behave like a high-end Web CMS (although to achieve high-end performance, you’ll have to do some work).

In terms of feature set, Plone outshines the more popular Joomla! (another open-source CMS with a tightly coupled architecture) by a significant margin, with its advanced workflow capabilities and potentially very powerful security model, not to mention in-context editing, an event-trigger system, automatic resource compression, and a pervasive object orientation, throughout. Plone’s careful attention to accessibility may pay impressive dividends.

Nevertheless, you pay a price for all of this wonderfulness, in terms of foundational technology that can seem at times to have the odor of flakeware (particularly if you come from a C++ or Java background). The problem isn’t that things like ZODB or “acquisitions algebra” don’t have demonstrable advantages or aren’t well proven in practice. Rather, it’s that they create an immediate learning-curve for the overwhelming majority of technology experts who know SQL but not Python; MySQL or SQL Server but not ZODB; URIs and URLs but not “acquisitions;” Apache but not Zope; and on and on.

Additionally, usability remains a concern. The product is open enough that you can improve some of the usability yourself (presuming you have Python and Zope skills), but the inscrutable ZMI is not something you’re going to re-chrome from bare metal.

The bottom line (as with any open source technology): Download and test-drive it, yourself. See what it takes to get a proof-of-concept working. You’ll know soon enough if Plone is the right fit for you.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

TYPO3: TYPO3

typo3.org

Project at a Glance

Specsheet	TYPO3: TYPO3 4.5 Summary
Geography	Global, with largest base in Europe (Germany)
What's New	Substantial UI modernization
Strengths	<ul style="list-style-type: none"> • Large and comparatively active community keeps project dynamic • UI localized in an impressive forty-two languages • Exceptionally modular architecture eases customization, integration, and replacement of services • Wide array of optional plugins broadens feature set • Good workflow compared to most open source alternatives • Various clustering options for high performance • Runs unusually well on Windows ("WAMP")
Weaknesses	<ul style="list-style-type: none"> • Out-of-the-box interface can be intimidating, especially for novices; very power-user oriented • Seems heavily reliant on optional modules of varying provenance, standards support, and accessibility • Developers must use obscure configuration language, "TypoScript," which has a steep learning curve • Templating requires a developer • Comparatively weak metadata support • Community-based development seems slow; version 5 has been in the works for more than three years • Community not as deep outside of Europe, particularly lacks breadth of system integrators in North America
Potential Fit	Informational Site, Community-Oriented Site, SMB/Departmental Digital Workplace
Unlikely Fit	Advanced Marketing Portal, Global Enterprise, Multichannel Publishing
Compare To	eZ Publish, Plone, OpenCms, Drupal
Operating Systems	Linux, UNIX, Mac, Windows
Repository	Any ODBC-compliant database: MySQL, Oracle, et al.
Client	Browser: Internet Explorer, Safari
App Platform	PHP 5.2+
Licensing	Open source: GPL
Ownership	TYPO3 Foundation

Summary

TYPO3 is a quite mature and extensible open source package built on the “LAMP” platform. The community is large and vibrant, but seems to be most active in Europe, primarily Germany. Architecturally, the package is a “coupled” management and delivery platform, although the two services are distinct enough to enable substantial modularization.

A major strength of the platform is its ready availability of numerous plugin modules. On the downside, the user interface remains somewhat dated and cluttered, even though version 4.5 is an improvement. Like many counterparts, TYPO3 in its present incarnation really favors power users over casual contributors.

The product is best suited to (and sees most implementations in) smaller government sites and/or interactive mid-sized sites, where there is a premium on corollary services, such as polls, forums, and blogs. TYPO3 can be made to scale nicely and might be suitable for an Ultra-Large Single Site, although pre-generating static HTML is not simple.

On the whole, TYPO3 seems to have lost some energy to Drupal, although it remains a decidedly community-based open source platform, as opposed to Drupal’s rather heavy reliance on its commercial sponsor, Acquia.

Scenario Fits	
Simpler Site	
Informational	<input checked="" type="radio"/>
Microsites & Landing	<input checked="" type="radio"/>
Mid-Range	
Basic Digital Marketing	<input checked="" type="radio"/>
Mobile Site	<input checked="" type="radio"/>
Community Oriented	<input checked="" type="radio"/>
Complex Site	
Advanced Marketing Portal	<input type="radio"/>
Global Enterprise	<input type="radio"/>
Multichannel Publishing	<input type="radio"/>
Ultra-Large Single	<input checked="" type="radio"/>
Digital Workplace	
SMB/Departmental	<input checked="" type="radio"/>
Global/Enterprise	<input checked="" type="radio"/>

Introduction

Development of TYPO3 was started in 1998, initially as a commercial product, until lead architect and programmer Kasper Skårhøj left his startup software company, Superfish, to focus on developing the application. Publishing the code under relatively loose, GPL terms — together with Skårhøj’s strong passion for the project — made for rapid uptake of the codebase.

Most implementations identify themselves as commercial sites for small- to medium-sized companies in Europe. With an active user base and a comparatively well-managed development community, TYPO3 has joined a handful of leading open source content management projects. It even sports its own (German-language) print magazine: *T3N* (<http://t3n.de/>).

The 2006 release of version 4.0 remains the current branch. This major version introduced some AJAX-powered features to improve the management interface, such as drag-and-drop movement of pages on the site hierarchy. TYPO3 used to be infamous for a jumble of tiny icons and incomprehensible dialog screens. Recent releases have updated editorial interfaces, removed the tiny icons and streamlined the workflow. While TYPO3 is most commonly used for its web content management capabilities, it is architected as a content management framework for other forms of content as well. Web content management is implemented as a

module on top of a framework that can be extended with other modules such as a file management system and a Digital Asset Management (DAM) system.

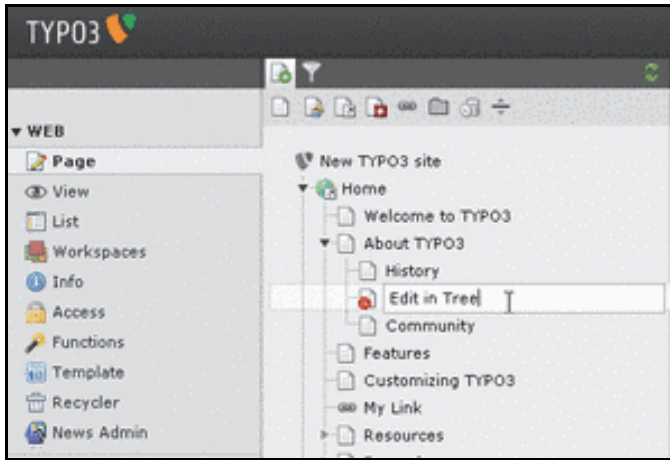


Figure 224. Much of the iconography and overall system layout has been modernized in V 4.5.

Version 5, code named “Phoenix,” has been pushed back repeatedly. An early alpha was expected for May 2009, then late 2009, and then fall 2010. The actual release date is still shrouded in mystery. One gets a sense that the version 5 development is sapping some of the energy out of the development of the current v4 branch. However, development there is benefiting from code developed for version 5, which accounts for at least some of the delay. Some

components of the “FLOW3” framework underlying version 5 have been back-ported to version 4 — specifically, some improved caching and plugin support. Rather than a big-bang scenario, the TYPO3 developers are transitioning from version 4 to 5 more gradually.

Technology

Technical Administration & Security

TYPO3 is built on the LAMP (Linux, Apache, MySQL, and PHP) platform. The package is optimized for MySQL although other databases are supported. It seems to install and run effectively on a “WAMP” (Windows/Apache/MySQL/PHP) platform, as well. Some websites run TYPO3 on Windows with the Microsoft IIS web server, although this configuration seems relatively rare.

TYPO3 is primarily a framework. It has been designed for modularity and extensibility, and those priorities are visible throughout the application. Other open source Web CMS tools, such as Drupal, WordPress, and Joomla!, are built to allow pluggable extended modules as well, but TYPO3 takes it to the next level. Each piece of contributor-facing functionality (including web content management, user management, and system administration functionality) is implemented as a module on the core framework.

As with Plone, the capabilities of the framework have managed to garner a solid following. However, as with any framework, actually making use of those capabilities requires in-depth knowledge of both the system specifically and how to apply content management best

Technology	
Technical Administration & Security	
Threat Prevalence	◐
Authentication & Authorization	◐
System Reporting	◐
Multisite Management	◐
Cloud Services	○
Development	
Configuration & Customization	◐
Integration & Extension	◐
Content Modeling	◐
Templating	◐
Performance	
Back-end Performance	◐
Site Caching & Delivery	◐

practices in general. Think carefully if you'd actually make use of some of TYPO3's more advanced concepts; a simpler product may prove to be more effective.

By default, TYPO3 publishes pages dynamically, but static publishing can be implemented. The "baked" option precludes the use of certain front-end dynamic plugins available with the platform. Note, however, that static publishing becomes a fairly complicated configuration, especially when you also require friendly URLs. There are a few add-on modules that attempt to simplify this configuration, but in reality, static publishing is rarely employed.

A fine-grained permission system controls access at the content level, and entire modules can be turned off for groups of users. In fact, with such a plethora of options (many of which are entirely TYPO3-specific), it's easy to get lost. As with many aspects of the system, it provides you a lot of flexibility, but it will take considerable experience to understand how to get things done.

Over the years, various administrative tools have been built into the back-end interface, but these are mostly for *technical* administration. The system comes with two reports: "Installed Services" and "Status Report", both of which are mostly useful for developers and administrators. There is also a comprehensive logfile viewer with various filtering options. However, this is also poorly suited for generating simple audit trails or user reports.

Development

Unlike most modern CMS (which typically employ standard scripting or tagging languages to display content), TYPO3 uses a proprietary language called TypoScript. Despite its name, it's not a scripting language. It doesn't have concepts like loops and other procedural control structures. TypoScript is totally declarative, only allowing you to set page attributes to various values. It's perhaps most comparable to XSLT, and it can be just as hard for developers to wrap their heads around it — but XSLT is a standard with broad acceptance. Learning TypoScript, by contrast, is not considered to be a great career builder.

Typically, TYPO3 developers use a static HTML template that provides the general layout of pages on the site. Areas for dynamic content are blocked out using special commenting syntax that is replaced by TypoScript-generated content. Extensions such as Auto-Parse ("modern template development") and TemplaVoila! ("futuristic template development") have tried to make it easier to adapt HTML designs into TYPO3 templates. However, this is still hard to learn for a developer, and it is likely to be unmanageable for a designer. Consequently, the process feels quite detached. You will need a developer to help your designers modify any templates once they reside in the system.

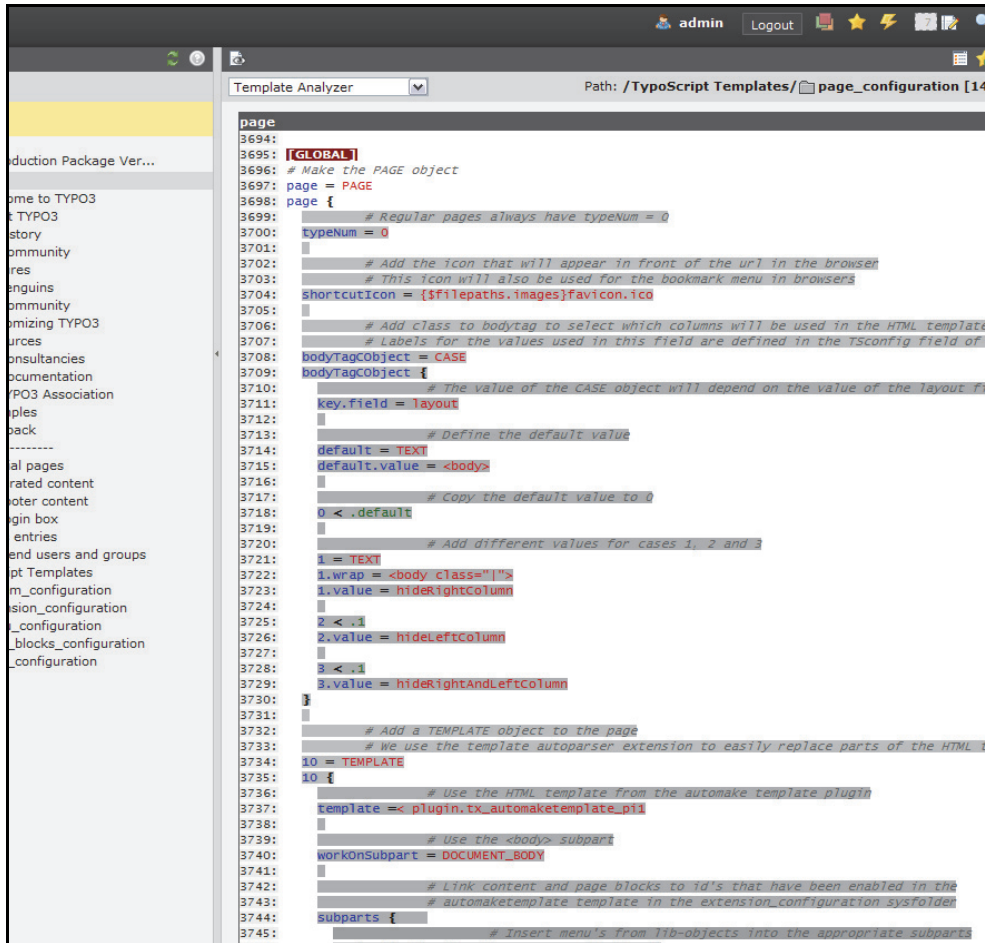


Figure 225. TypoScript configuration code in the back-end interface.

The “Liquid” templating language is slowly making this easier. It was developed for the future version 5, and (like other parts of that branch) it has been “backported” to the version 4 branch. It’s an altogether more modern way of templating using simplified tag markup and clean functions. The Liquid tags are much easier to maintain in an external HTML editor (say, DreamWeaver) than TYPO3’s troublesome comment blocks. Additionally, the language was designed to be extendable with “ViewHelpers.” Fluid templates will be future-proof for version 5. However, the backport was first introduced in version 4.3, and isn’t fully functional yet; experience with it remains somewhat limited.

Web content management in TYPO3 is focused on two high-level types: *Pages*, and *Content Elements*. Pages are placed in a hierarchical tree structure that represents the navigation of the site. Pages are easily moved by dragging and dropping icons on the page tree. Links are not broken because all URLs are referenced by an ID, rather than a path. Of course, the consequence of this architecture is that URLs are less intuitive — a limitation that is addressed with the extension, “RealURL.”

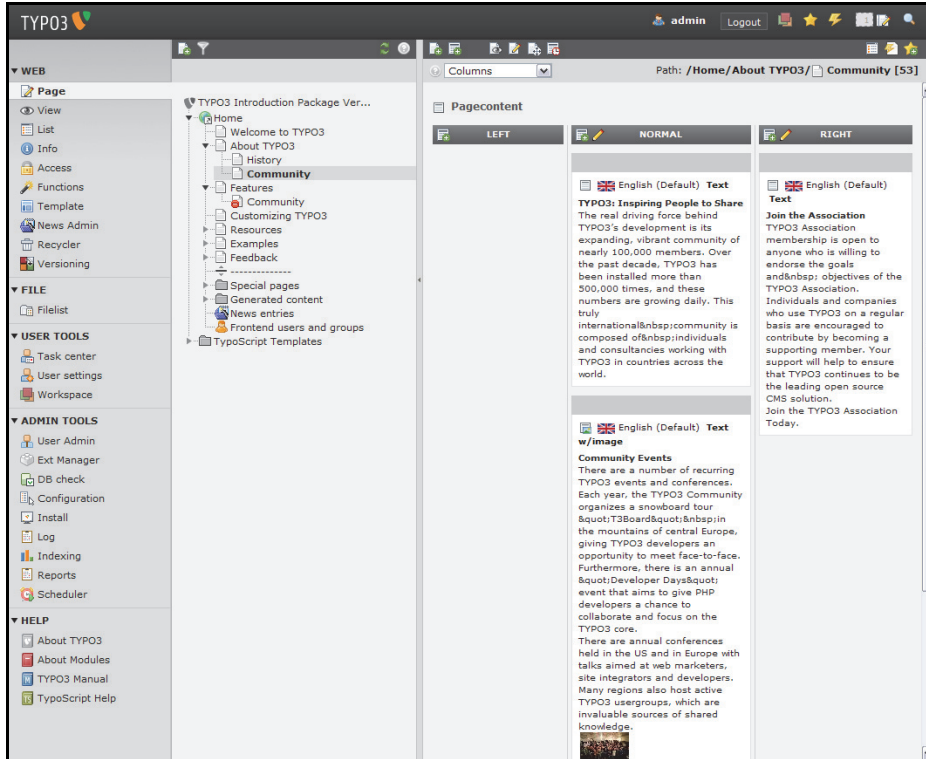


Figure 226. “Pages” in the tree contain “Content Elements,” which are grouped in columns.

Content Elements are considered chunks of a page (and part of a specific column). Although the content model is component-oriented, TYPO3 is really a page-based system. It’s possible to reuse Content Elements on other pages, but editors will have to “insert a record” and search for the element in the repository.

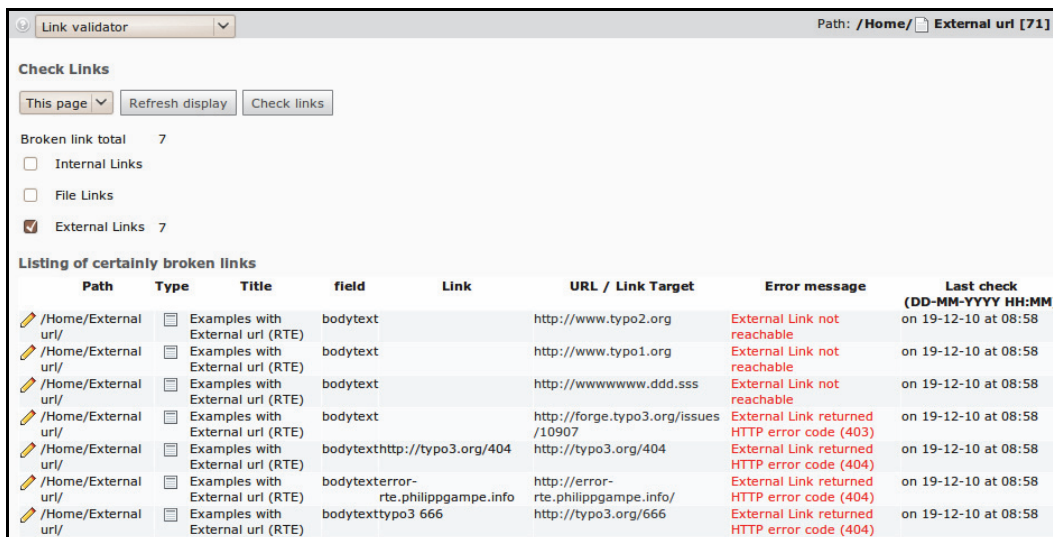


Figure 227. The link checker can check both internal and external links.

Because formatting is closely tied to the element, it will then display in its original design, whether or not it suits the new page. Copying and pasting elements is quite easy though, and more importantly, intelligible to users. This means TYPO3 is fine for repurposing, but not well suited for content reuse.

Performance

While its LAMP architecture makes it possible to run on inexpensive hardware, the application is relatively resource intensive due to the layers of abstraction and complexity built into the architecture. Whereas simpler PHP-based WCM systems are customized by modifying or writing PHP code, TYPO3 has implemented a whole new configuration language that needs to be parsed and interpreted by PHP which is an interpreted language. The flexibility of the database design has a performance cost, as well. The impact of this is that low-end, shared hosting plans are impractical for most sites. Still TYPO3 is less resource hungry than many CMS products written in Java, and the editorial interface performance is comparatively speedy.

High-traffic websites can employ various clustering configurations using one or more instances of the application and one or more databases. The most common method is to have multiple servers running the TYPO3 application pointing to the same database server. There are known installations with ten Apache web servers all pointing to the same MySQL database server. Another commonly used configuration is to run multiple database servers. This option, while providing better redundancy for failover, is more complex to set up, because it involves database replication.

Content

Contributor Experience

The back-end editorial interface is a lot to digest for most non-technical users. Version 4.5 introduced an improved back-end interface, in part by implementing ExtJS (JavaScript libraries that support AJAX and DHTML) components to support features like drag and drop. The new interface also replaces the old help icons with genuine help information.

What used to be one of the most chaotic interfaces of any CMS has greatly improved since the clutter of version 3. Nevertheless, it remains a power-user oriented system, and TYPO3 is still far from intuitive for a novice user.

The left side of the management interface lists (based on ExtJS) the modules that are installed and available to the user. Selecting a module on the left brings up the administrative interface for that module. Modules needed for most web content management applications come pre-installed with the core distribution. In version 4.4, one of these modules was the revised “Task center,” which was designed to be the user dashboard. Unfortunately (and rather typically), the

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

documentation is nowhere to be found, and out of the box, it lacks any meaningful functionality.

The complexity of the back-end is somewhat mitigated by TYPO3's in-context editing capability where authenticated and authorized contributors see an administration panel which allows them to edit content and execute various content management functions such as workflow and publishing. In some successful implementations of TYPO3, casual users can completely avoid the back-end.

Ultimately, TYPO3 remains a long way off from understanding how improve its usability for the non-technical user. Few editors would ever voluntarily pick TYPO3 over WordPress. Even Drupal (hardly an icon of usability itself) is probably easier for the casual editor to use.

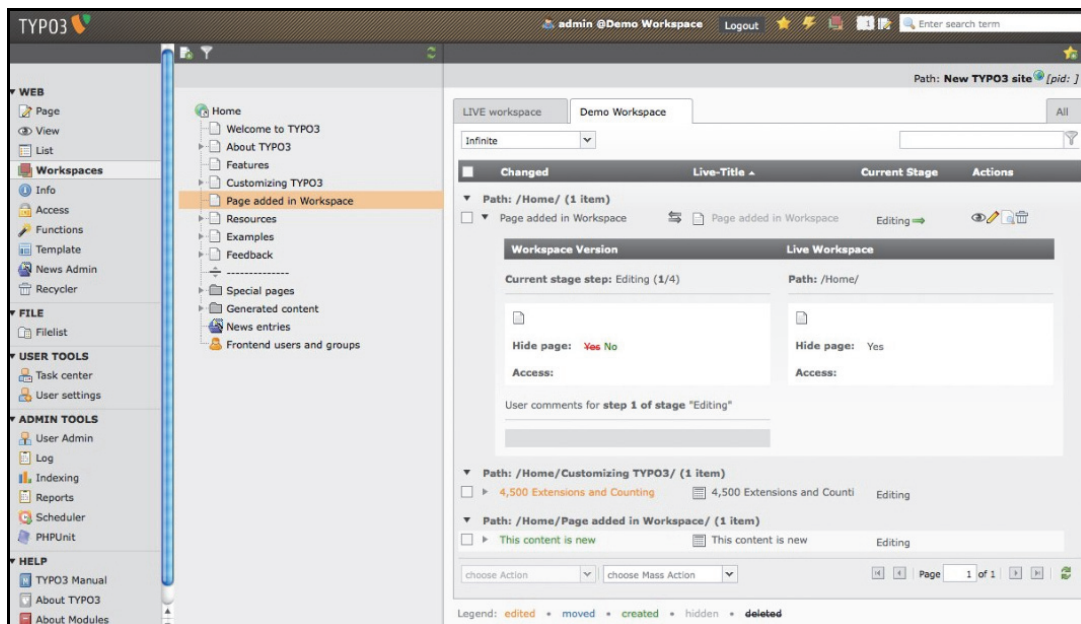


Figure 228. The editorial dashboard has been streamlined, but when fully realized it can be complicated and requires careful training.

The TYPO3 back-end is localized with language packs for about 40 languages. Database storage can handle extended character sets. Most modules come with language pack files for translating messages and labels. Similarly, presentation templates can have tokens, which are replaced by translation lists. There is a developer's tool in an extension called ExtDevEval, which extracts hard-coded text strings from code, puts them in a language file, and replaces the text with a translation key reference to the language file.

Contributing Content

TYPO3 ships with a default layout that employs four possible columns. These can be changed by either disabling columns or more in-depth customization. However, for most page layouts, the column framework is effective. There are several default content element types, each of which can represent a block of text on a page. In addition to basic types (such as text, and text with image and tables), there are also elements for multimedia, forms, and visitor-facing plugins.

This semi-structured model is a reasonable compromise between content definitions that give users a full page to fill with unstructured markup, and more structured, decomposable content types (such as a “review,” which may have fields for item reviewed, score, etc.). The trade-off between these two extremes is flexibility and ease of use, versus content reusability. A content element, while reusable for inserting into other pages, is based on its layout rather than its semantic meaning, so it would be difficult to display differently. For example, the primary content element types are syntactical ones like *paragraph*, *paragraph with image*, and *table*, not format-neutral, semantic element types such as *summary*. Through CSS, the visual treatment (or branding) of these content blocks can be manipulated, but only to a certain extent. Out of box, TYPO3 does not support custom content types, but you can create them via other optional modules, some of which have front-end plugins for placement in pages within the web module.

Most content element types have a WYSIWYG editor based on the open source project `htmlArea`. `htmlArea` has gone through a bit of a transition with most its development by switching over to the “Xinha” project. TYPO3 so far has stuck it out with `htmlArea`, integrating it in the new ExtJS-based interface. With ExtJS also changing into the Sencha commercial open source project, it’s fair to say that TYPO3 hasn’t had the best of luck in its choices of technologies.

You’ll find a sophisticated versioning system that allows full-site versioning, which is incorporated into the core user interface. Using the versioning feature, a user can roll back an entire page. It is also possible to preview the entire workspace; the preview has a unique URL that enables it to be viewed without logging in. This is useful when asking third parties (that don’t have an account in your system) to comment on the site you’re working on; you can simply email them the URL and they can have a look. The downside, of course, is that this can have serious security implications when dealing with sensitive content, which won’t always be apparent. (Systems containing somewhat similar functionality, such as Google Apps, have a history of recipients thoughtlessly forwarding the preview link.)

If you need to schedule publishing, you can configure a start and end date for new content to be visible. Perhaps it is a bit simplistic by default, but some users have decided to build on this to meet their requirements, e.g., for publishing on a specific date and time.

Metadata support is not particularly strong at either the Page or the Content Element level, with neither having extended properties such as tags or categories (as you find in Drupal or WordPress). The TYPO3 web module only tracks information that is used for management purposes — such as site hierarchy placement. However, there are several modules that can extend the basic web module with metadata elements.

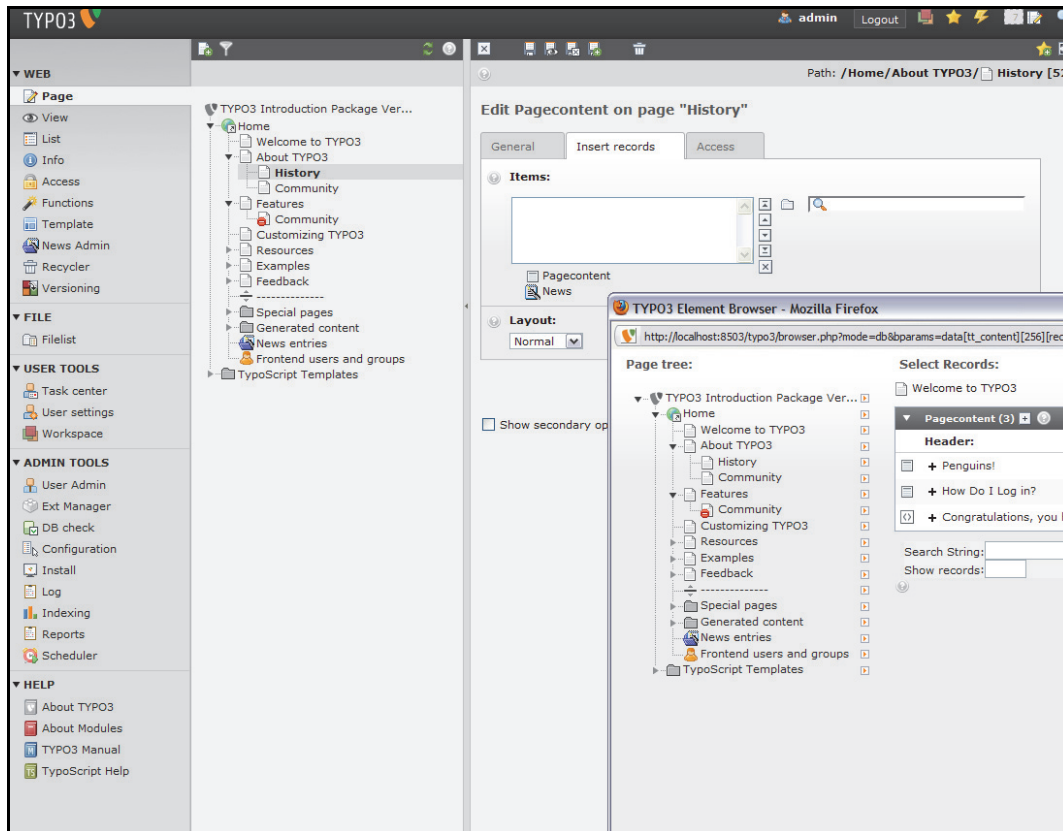


Figure 229. Contributors can reuse content components via an “Element Browser” pop-up. Note the number of panels and icons in the default interface.

The TYPO3 core has sparse functionality for multimedia. It’s possible to insert various types of content into a Page, but there are few ways to *manage* the assets. Various extensions improve on this; however, even an extension like Media (DAM) is still limited compared to what most systems offer as core functionality.

Content Lifecycle

Workflow management in TYPO3 is handled via “Workspaces,” which are a potentially very powerful and flexible abstraction. As with many aspects of TYPO3, however, the power has been hard to harness and most users stick to simple staging/live scenarios. With release 4.5, workspaces have at least received some usability attention, with screens becoming more logical and easier for users to navigate.

Combined with the workflow module, a content manager can work in a workspace where no edits are visible on the live site until that workspace has been published. The publishing process is executed by a button and takes effect immediately. However, each page has properties for start and end time, which identify when the page is visible. Using this feature, an administrator can publish content immediately, but will not have it visible until a specified date; there are tools to visualize what changes have been made to the workspace. Note that publishing occurs at the workspace level, so it is not possible to hold back a set of changes when publishing. Obviously, you’ll need to plan carefully here. Note that modifications to

other managed objects (such as templates) can also be handled via workspaces, which could be quite useful for developer-designer cooperation.

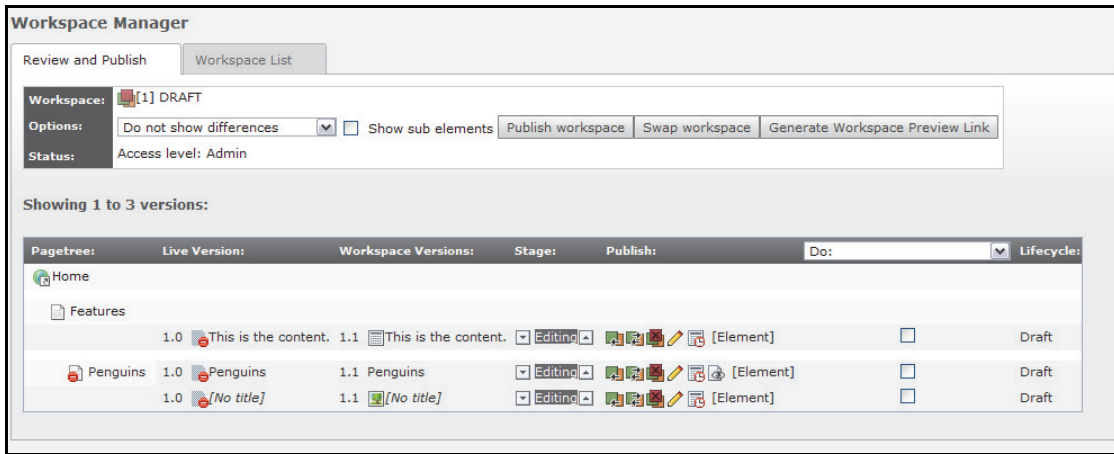


Figure 230. The “Workspace” concept is potentially powerful and flexible, but it also can be hard to understand and use.

There are two predominant methods for creating multiple language versions of sites in TYPO3: the “Two Tree” approach, where the two localizations are managed independently as individual sites, and the “One Tree” approach, where you create translations of individual pages in a common tree. The latter is used less frequently, and requires an extension called “Language Selection,” but it has the advantage of allowing for workflows, with new content that is automatically dispatched for translation.

Experience

Digital Marketing

Natively, the product does not always output clean, friendly HTML. In particular, you may need to remove table-based markup.

With the default publishing, TYPO3 uses the typical “index.php?id=256” URLs many PHP systems employ. For friendly URLs, you must install an extension.

Ancillary Services

TYPO3 has a native site search engine that indexes page content and documents, the latter of which is based on the OASIS OpenDocument format. If you want to use an external search crawler (such as a Google Appliance), it is recommended to use the extension RealURL, which converts query string-based URLs to more logical URL paths. RealURL works with the Apache mod_rewrite module.

The receptiveness of the architecture, combined with the energy of the community, has made a thriving exchange of TYPO3 extensions. There is a rich and well-documented API for PHP programmers to extend the application with extensions that can be used on either the front-end or back-end. As with other open source projects, documentation for extensions ranges from nonexistent to thorough, with the more popular extensions having manuals and articles explaining configuration and usage. Through the back-end, an administrator is able to search, import, and install extensions hosted on the TYPO3.org website. Just beware that not all extensions play nicely with each other: test, test, and test.

Experience	
Publishing	
Standards Adherence	
Multichannel	
Mobile	
Digital Marketing	
Site & Campaign Analytics	
Testing & Optimization	
Segmentation & Personalization	
Social Media Integration	
Promotional Campaigns	
Community & UCG	
Workplace	
Collaboration & Networking	
Dashboard	
Ancillary	
Site Search	
Online Forms	
Module Ecosystem	

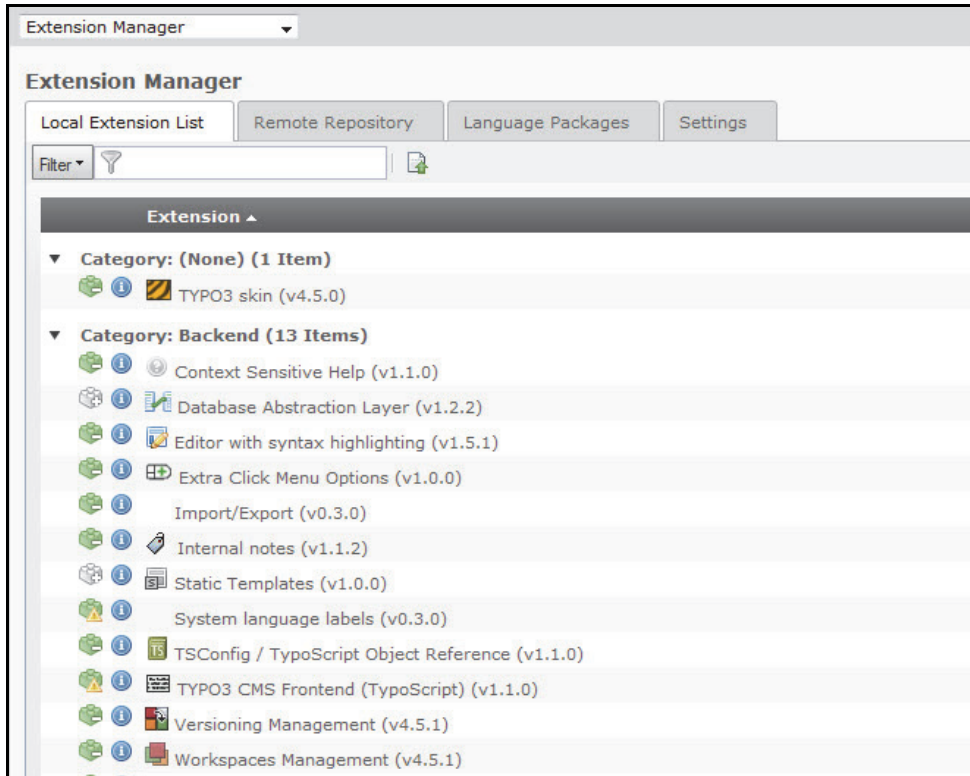


Figure 231. The Extension Manager in v4.5.

Project Intangibles

In Europe, the TYPO3 community is large, active, and well organized, especially in Germany. Beyond individual users and developers, there is also a broad base of consultancies and hosting companies specializing in providing services around TYPO3. Outside of Europe (e.g., North America), experienced consultants can be rare — though you can find them if you dig. According to TYPO3 users, the many system integrators have grown increasingly experienced. Nevertheless, if you are a large, global organization — meaning not a typical TYPO3 user — you’ll still have a hard time finding experienced help.

A non-profit organization, the TYPO3 Association, supports the core development of the software and serves as a management body for the community. The TYPO3 Association functions to provide stability and continuity like its peers (such as the Apache Software Foundation), but it has some added nuances. The TYPO3 Association has two levels of membership: Active and Supporting. Active members are nominated based on their contributions to the project and have the right to vote on project matters. Supporting membership is open to anyone, but requires an application process and membership fees. The Supporting Member program is somewhat analogous to a paid partnership program, but also includes TYPO3 customers, as well as consultancies.

Intangibles	
Vendor Services	
Vendor Professional Services	<input type="radio"/>
Channel Partner Services	<input type="radio"/>
Support & Community	<input type="radio"/>
Strategy & Roadmap	<input type="radio"/>
Viability & Stability	<input type="radio"/>

The TYPO3 Association started a consultant certification program in 2008. Note that although the official title is “Certified TYPO3 Integrator,” the process consists of testing individuals (people can become “certified integrators,” not companies). The Association says there are now over 600 certified consultants, with over half of them in Germany and only one in the US. There is a separate list of registered service providers that includes a count of how many sites each company has built. The Association has been working on a certification process for these companies for some time, but for now lists “approved consultancies” without further rationale.

You’ll find many contributed extensions listed on the TYPO3.org site, and the breadth of these modules is frequently cited as a boon to customers. Unlike most open source projects, TYPO3 has a group dedicated to evaluating and rating the quality and utility of these extensions, but only a small subset has been formally reviewed. Note, however, that extensions aren’t reviewed for interoperability; even certified modules are not guaranteed to play nicely together.

The prolific English-language mailing list has a digest mode that most users prefer, due to the volume of messages sent each day. Questions to the list usually receive a response within hours. The IRC channel is also a good source for support.

TYPO3 boasts an expansive library of documentation, and several books have been published (although most of them are in German). However, most of the documentation is outdated and a few steps behind the version 5 to version 4 retrofits (see below). Summed up, one developer who completed a complex TYPO3 implementation said, “TYPO3 documentation is bad, but compared to other open source projects, there’s more of it.”

Development of the version 4 branch continues in parallel with work on the future version 5, which includes a complete redesign of the technology while retaining most of the current interface. The 5.0 development team promises to bring modern development principles to the PHP community. (Version 4, while compatible with PHP 5, does not take advantage of its new features.) Ambitiously, part of the project is building a content repository in PHP according to the Java content repository specification called “phpCR.”

Originally, the plan was for TYPO3 5.0 to be a complete departure from previous versions, sacrificing backwards compatibility and starting from scratch. However, at the end of 2008, the version 4 and version 5 developer teams met up and decided to integrate their efforts more tightly. This led to what the community has dubbed “the Berlin Manifesto.” Development of v4 continues, but the aim is to merge features between the two versions.

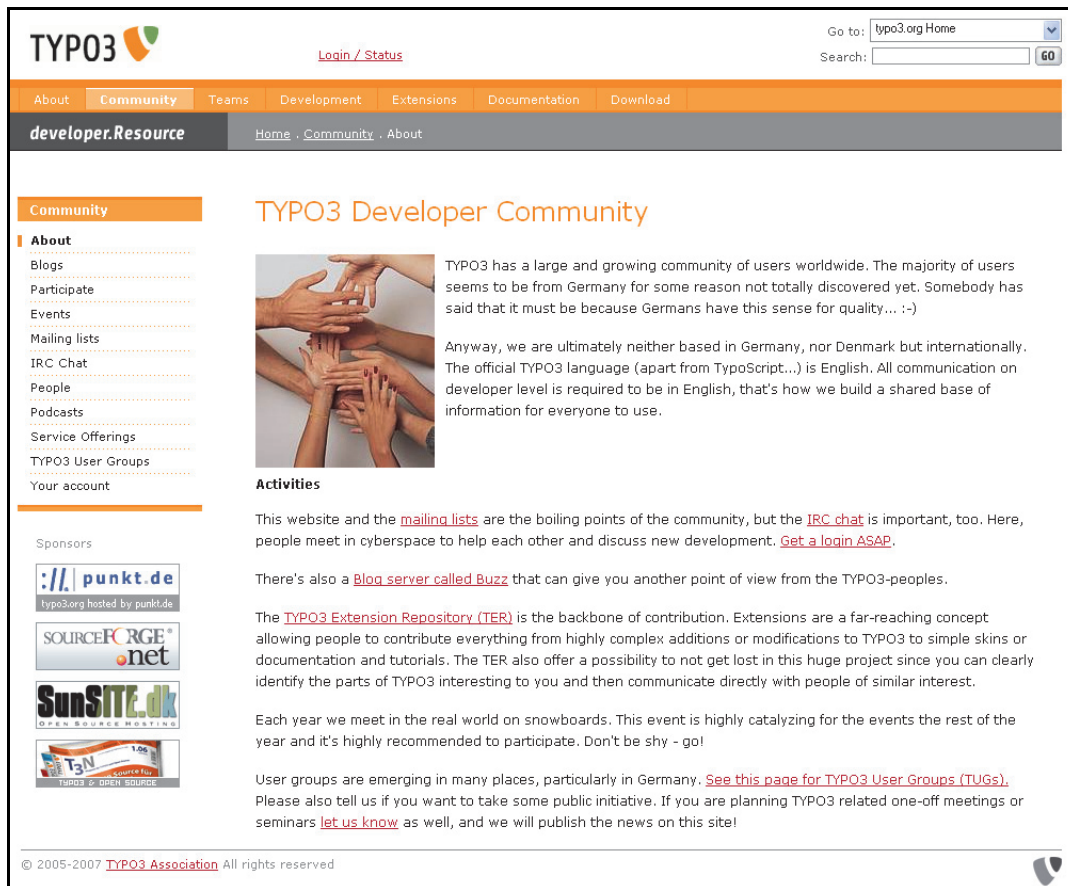


Figure 232. The TYPO3 Developer Community.

In practice, this means that version 5's features are "backported". The caching and plugin framework from FLOW3, ExtBase, and the templating language Fluid are the first examples. Hopefully, this means that plugins developed for 4.4 should also work in 5.0. The developers have promised to work on an easy upgrade path, as well. However, make no mistake; the architecture is such a complete departure from the earlier versions, that an "easy" migration will still be a daunting task.

Conclusion

Quirks aside, the TYPO3 framework has a long history of quality and stability. Its list of registered sites is impressively long (more than 500K public-facing sites), and shows the range of possibilities of what the product can do. In the current version, TYPO3 has been somewhat behind the curve, in terms of contributor usability, and standards-based markup, but it made up for its shortcomings with its flexibility. In the hands of a seasoned developer, the system is capable of some impressive feats.

However, the project is now in the twilight zone between the proprietary technology and design choices of version 4 and the off-the-beaten-path choices of version 5. This puts a strain on TYPO3 veterans to keep up, and it is not exactly an open invitation for new talent. Until then, you'll have to figure out whether to stick with the aging AutoParse or TemplaVoila!

templating, or to go with the new and unproven Liquid. New minor releases are likely to bring equally difficult choices between the Ghost of TYPO3 Past and the Ghost of TYPO3 Future. In either case, you must carefully identify and test modules — and interoperability — before committing to the platform.

Most importantly, the system's user interface is not for the faint of heart. It has come a long way, and it has seriously improved. In all likelihood, however, it may still come off as too clunky. Test very carefully here — with real users — before you commit.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Mid-Range Product Vendors: Roll-Up Comparisons

- CrownPeak: CrownPeak CMS
- Ektron: Ektron CMS
- Enonic: CMS
- e-Spirit: FirstSpirit
- eZ Systems: eZ Publish Platform
- Hannon Hill: Cascade Server
- Ingeniux: Content Management System
- Upland Software: Clickability
- Magnolia: Magnolia
- CCI: Escenic

A key advantage of this class of products as a whole is that they tend to offer more pre-packaged features. Many of them actively promote “website-in-a-box” solutions that include bundled social applications, analytics, experience management, and so forth. This is particularly attractive if you are starting a new site from scratch and don’t have pesky legacy content or preexisting applications to address. They also tend to require fewer development resources.

Note that you are trading off against potential customization and scalability limitations, and that product APIs almost certainly will not run as deep as those in the “platform” tiers. You can customize these tools to a point, but then you’re very much into uncharted territory.

These packages typically carry initial price tags of US \$20,000–80,000, but as always, the final fee can creep upward, depending on the final number of servers required or contributors in the system. Therefore, it pays to know up front if you are going to need clearly delineated development, staging, and production environments. Development and staging license fees are often negotiable.

Specsheet Summary: Mid-range Products

	CrownPeak	Ektron	Enonic
Geography	North America and Europe	Global	Primarily Norway and the rest of Scandinavia
What's New	<ul style="list-style-type: none"> Integration with the Demandbase targeting platform for personalization Major overhaul of the user interface 	<ul style="list-style-type: none"> Aloha as default rich text editor in lieu of Ektron's own eWebEdit400 Editor Cloud deployment options on Windows Azure and Amazon EC2 Enhanced SharePoint integration 	<ul style="list-style-type: none"> Improved versioning and page "snapshot" capabilities Disk-based binary store replaced database storage System performance tracing tool Friendly URLs generate as standard "Assign" workflow feature replaces previous one-step system
Strengths	<ul style="list-style-type: none"> Longest-standing SaaS-based CMS Company is finding a niche in multisite management Comparatively easy to roll out new sites, including microsites Globalization features are supported through integrations and translation workflows Boasts some major intranet deployments Strong and consultative customer support Developers feel comfortable in the .NET habitat 	<ul style="list-style-type: none"> Broad range of out-of-the-box features, with special emphasis on fast-evolving social tooling Support for reusable widgets (.NET controls) gives web designers flexibility in creating complex pages quickly eSync module enables content deployment across environments, round-trip synchronization, and other related deployment tasks Unlike Sitecore and EPiServer, company sells products and consulting directly to customers Large, active developer community Customers report good first-line product support 	<ul style="list-style-type: none"> Leverages standard J2EE technologies familiar to most Java developers Broader-than-average support for databases and appservers SOAP-RPC API, in addition to pure-Java API Relatively low learning curve for authors and approvers Good value: decent feature set for the price Company and product very straightforward: the antithesis of "slick"

(Continued)	CrownPeak	Ektron	Enonic
Weaknesses	<ul style="list-style-type: none"> • Default interfaces favor power users and may need “dumbing down” • UI is English only • Historical lack of web experience management tools (e.g., campaign management and social analytics), makes it less suited for online marketing use cases • Decoupled architecture tends to mitigate against more interactive, personalized scenarios • New C# API is generally only used by new customers thus far • Dearth of collaboration, mobile, and social services • Page-oriented system does not encourage content reuse • Underdeveloped community hinders customer cooperation • Corporate strategy and development has a tendency to meander • Future of the company is somewhat uncertain in light of aging venture investment that may push a premature “exit” 	<ul style="list-style-type: none"> • Ektron tends to do things its own way, rather than via industry norms • Comparatively limited and not user-friendly multilingual capabilities • LDAP integration lacks depth, Single Sign-On (SSO) capabilities are not available • Setup, deployment, configuration, and templating can be complicated; architecture is older and less standardized than that of some competitors • Emphasis on packaged features comes at the expense of developer extensibility, relative to its direct competition • Complaints of channel cannibalization have led some consulting partners to re-assess their involvement with Ektron • Ektron has a history of releasing under-tested code in its rush to add more features 	<ul style="list-style-type: none"> • Highly XSLT-centric, which means a steep learning curve • Cursory documentation awaits developers who wade into system • Permissions management is potentially tedious • UI is oriented toward techie contributors rather than marketers • Coupling between menu and documents is complex • Workflow capabilities are primarily ad hoc in nature • Surprising native absence of support for globalization scenarios • No integration with Microsoft Office • No out-of-the-box blogging, wiki, or collaboration apps • Not well-known outside of Norway
Potential Fit	Informational, Micro-sites & Landing, SMB/ Departmental Digital Workplace	Basic Digital Marketing, Microsites & Landing Pages, Community Oriented, SMB/Departmental Digital Workplace	Microsites & Landing, Multichannel Publishing, Basic Digital Marketing
Unlikely Fit	Ultra-Large Single, Advanced Marketing Portal, Multichannel Publishing, Community Oriented	Advanced Marketing Portal, Multichannel Publishing, Global Enterprise, Ultra-Large Single Site	Advanced Marketing Portal, Digital Workplace, Global Enterprise, Ultra-Large Single Site
Compare To	Upland Software, Ektron, Sitecore, EPiServer	Sitecore, Kentico, EPiServer, Telerik, SDL	CoreMedia, EPiServer, Escenic, Oracle, Magnolia, e-Spirit
Operating Systems	N/A	Windows	IBM AIX, Windows, Linux
Repository	N/A	Database: Microsoft SQL Server	Database: Most RDBMSs supported, including MySQL 5.1 and later
Client	Any browser supporting Silverlight	Browser: IE, Chrome, Safari, and Firefox on Win, Mac, and Linux	Browser
App Platform	Hosted platform with .NET, SOAP/REST APIs	.NET	Java application server: Tomcat, BEA WebLogic, IBM WebSphere, JBoss, Oracle Application Server, Sun AS, Glassfish

(Continued)	CrownPeak	Ektron	Enonic
Licensing	Based on the size of repository and the number of contributors, it starts at US\$72K annually, exclusive of implementation and support options	Combination of URLs and contributors. 1 URL/5 users edition starts at US\$15K, Enterprise Edition \$100K and up. Add-on modules sold at extra 20% of the base license fee each.	Enterprise starting at €16,000. Community Edition under open source AGPL 3.0 license
Ownership	Privately held, VC funded	Privately held	Private — majority owned by management and private investors

	e-Spirit	eZ Publish Platform	Hannon Hill
Geography	Primarily Europe, some in North America	Global, with a major footprint in Western Europe	Primarily North America
What's New	<ul style="list-style-type: none"> • New HTML5 browser client • Re-written thick Java client • Optional module for front-end content integrations 	<ul style="list-style-type: none"> • Major architectural overhaul of the CMS • Embedded Symfony 2 web application framework as product's underpinnings • Rewrite of the public (PHP) API and REST-ful APIs 	<ul style="list-style-type: none"> • Enhancements to site publishing and previewing capabilities • Connectors to extend Cascade Server via third-party applications
Strengths	<ul style="list-style-type: none"> • Simplified WYSIWYG editing, without unnecessary features cluttering the interface • Rich editorial functionality offered by Java application-based thick client • Virtual "time machine" allows you to roll back site content arbitrarily • Strong integration with SAP NetWeaver Portal, WebSphere Portal, Liferay, and Microsoft SharePoint • Facilitates integration of external structured sources (such as databases) into websites • Offers good workflow • Codebase is traditionally quite stable • Decoupled architecture will appeal to some enterprises 	<ul style="list-style-type: none"> • Simple to install • Templating is comparatively easy to learn due to more adherence to a standards-compliant industry framework • Flexible and extensible development framework • Strong support for multimedia in general, and video in particular • eZ Find extension for internal search engine is based on Lucene/Solr • Comparatively easy to use in-context editing interface • Established, growing, and supportive community • Backed by a commercial company and VC funding 	<ul style="list-style-type: none"> • Product offers a straightforward, no-frills, "back-to-basics" approach to web content management • System was designed with content reuse (at the level of XHTML chunks) in mind • Relatively clean Java-based architecture • Comparatively strong support for rendering content in multiple output formats for multiple target devices • Velocity templating technology (already familiar to some Java programmers) can be used in place of harder-to-master XSLT • Site Import functionality aids in migrating existing content • Online developer knowledge base is extensive and well organized • Company is small and accessible, priding itself on taking a "high-touch" approach with customers. The company has good experience in higher education and government sectors

(Continued)	e-Spirit	eZ Publish Platform	Hannon Hill
Weaknesses	<ul style="list-style-type: none"> • Browser-based thin client only can be used for content editing; for some other tasks, you need the thick client (applet) • WYSIWYG editing is not Word-like • Relatively inflexible site structures hinders more advanced globalization efforts • Lack of key audit trails and reports reduces enterprise value • Does not leverage mainstream Java standards — uses several arcane choices for some subsystems, including templating • Initial product implementation can become quite complex • Comparatively pricey for what you get • The product's highly modular nature makes it hard to determine elements of the core platform 	<ul style="list-style-type: none"> • Local translations vary in quality, and globalization functionality remains incomplete • Scalability and performance issues traditionally haunt larger eZ implementations • Limited eZ Find search utility adds Java to PHP-based environment • Documentation (while improved) can be spotty • Administrative UI is difficult to use • Scarcity of tangible resources and support in North America • Continued turbulent internal changes on the executive levels • Online marketing capabilities are brand new and lack maturity 	<ul style="list-style-type: none"> • Very limited support for scenarios involving interactive and/or personalized content • Programmer heavy, with limited customization capabilities for business users • Customization requires extensive knowledge of XML, XSLT, Velocity, and frequently Java • Templating subsystem feels unnecessarily complex • Lack of proper staging mechanism makes the product less suitable for larger enterprise customers • Lacks ready-made "social software" applications; social connectors only support one-way integration with Twitter & WordPress • Comparatively little market penetration or domain expertise outside of higher education, government, and health care • Relatively few experienced integration partners
Potential Fit	Global Enterprise, Ultra-Large Single Site, Global/Enterprise Digital Workplace	Basic Digital Marketing, Informational Site, Multichannel Publishing	Informational Site
Unlikely Fit	Informational Site, Microsites & Landing Pages	Microsites & Landing Pages, Community-Oriented Site, Global/Enterprise Digital Workplace	Advanced Marketing Portal, Ultra-Large Single Site
Compare To	CoreMedia, Adobe, Escenic, Magnolia	Ektron, TYPO3, Drupal, Plone, EPiServer	Ektron, Ingeniux, OpenCms, OmniUpdate
Operating Systems	IBM AIX, Linux, Microsoft Windows, Sun Solaris	Windows, Linux, Macintosh, Solaris, FreeBSD, HP-UX	Windows, Mac, Solaris, Linux
Repository	Databases: IBM DB2, Microsoft SQL Server, MySQL, Oracle, PostgreSQL	Databases: Microsoft SQL, MySQL (on demand only), PostgreSQL, Oracle	Databases: MySQL 5.0/5.1, MS SQL Server 2005/2008, Oracle 10g/11g
Client	Any browser for content editors: Java client (Java WebStart) for template designer, administrators, and super users	Browser: Chrome, Safari, Firefox, IE, Opera	Browser: Internet Explorer 7+, Firefox 2+, Safari 3+, Chrome
App Platform	Java container: Supports IBM WebSphere, Tomcat, and Jetty	PHP5	Java 5; Cascade Server is bundled with Apache Tomcat
Licensing	Starts at €80,000, for a base license that covers unlimited CPUs and servers. Additional feature modules are sold separately.	Dual: Free GPL or proprietary licenses, average deal size US\$40K	US \$60,000 for single server, unlimited URLs, or \$40,000 for single CPU
Ownership	Privately held, owned entirely by consulting company	Privately held	Privately held

	Ingeniux	Upland Software/Clickability	Magnolia	CCI (Escenic)
Geography	Primarily North America	Primarily North America	Mostly EMEA, some North America and Asia-Pac	Primarily Europe, especially the Nordic region
What's New	<ul style="list-style-type: none"> • Support for in-context editing • Ability to create workflows visually • Single cross-browser client for all tasks 	<ul style="list-style-type: none"> • Single sign-on via SAML support • Integration with Demandbase for visitor profiling and targeting 	<ul style="list-style-type: none"> • Single interface for touchscreen and desktop editors • Personalization/segmentation module • API for integrating external media assets 	<ul style="list-style-type: none"> • New video workflow in Content Studio • New JavaScript framework for building extensions to Content Studio
Strengths	<ul style="list-style-type: none"> • Emphasis on structured content and XML makes Ingeniux well suited to repurposing content to multiple formats or channels • Supports a variety of configurations (coupled, decoupled, n-tier) and publishing styles (fried, baked, or mixed) • Only one contributor UI for all platforms (Safari, IE, and Firefox) • Supports cascaded authentication enabling simultaneous authentication for identity providers • Simple, page-oriented approach appeals to non-technical end users • Search-engine friendly URLs • Company has a reputation for providing good support 	<ul style="list-style-type: none"> • Company traditionally targeted functionality for media sector • Only major SaaS vendor to tightly couple content production and delivery, easing deployment, preview, and dynamic delivery • Comparatively broad site-management capabilities and user-generated content services, out of the box • Large tag library for customization by a reasonably technical webmaster, plus real development and staging environments • Most customers find the support team to be responsive and capable • Company's consulting team also held in very high regard • Impressive customer community has emerged for peer support • Redundant data centers offer greater reliability and continuity 	<ul style="list-style-type: none"> • Decent editorial features and comparatively friendly contributor UI • Comparatively open architecture and support for multiple templating engines makes the product easier to extend • Localized in an impressive 16 languages • Strong repository services helps support very large sites • Company focused on standards adherence • System and company are comparatively developer friendly, and tech support is unusually well regarded • Optional "Enterprise" edition brings more functionality for a fee 	<ul style="list-style-type: none"> • Broad support matrix • Strong fit for media companies, notably newspapers (where Escenic has developed substantial domain expertise) • Java-based architecture with extensive tag-library API • Content Studio thick client can be highly usable for power authors • Cross-media approach simplifies content reuse to multiple channels • Good support for basic approval workflows • Strong metatagging functionality • Many micro-application expansion modules available

(Continued)	Ingeniux	Upland Software/Clickability	Magnolia	CCI (Escenic)
Weaknesses	<ul style="list-style-type: none"> Search performance can be poor on large sites The collaboration module (Cartella) has features not present in the CMS Templating still requires coding and lacks a visual or drag-and-drop mechanism to create layouts Working with content types requires developers Pace of CMS product development has traditionally been sluggish Company is very small, with limited community, and has no presence outside North America Emphasis on XML and structured content is overkill for simpler scenarios Not a suitable choice in a non-Windows or mixed environment 	<ul style="list-style-type: none"> In this coupled SaaS CMS, you will want to monitor performance and cost carefully; it's architecturally too coupled for some large enterprises Comparatively burdensome integration and extension capabilities Interactive applications lag in comparison to best-of-breed alternatives Weak content lifecycle management Overall approach is page oriented, not component oriented Extremely limited digital asset management services Poor user and group management capabilities English-only UI — product interface is not internationalized 	<ul style="list-style-type: none"> Platform is very page oriented; not well suited to content reuse or even simple, metadata-based index pages Workflow is rather clunky and requires developer to model Surprising lack of services to support global publishing and distribution Thin asset services mitigates against media-heavy sites Very few built-in capabilities to support digital marketers, including a dearth of connectors to third-party marketing platforms Thorough understanding of Java and product-specific expertise required; it is easy to make poor technical decisions Community is sparse in some areas of the globe; It can be very difficult to find experienced specialists in North America 	<ul style="list-style-type: none"> For enterprises not in the media vertical, the learning curve will be steep and additional implementation support may be required Feature set is more broad than deep Workflow is a bit limited out of the box for enterprise use Product is split into a profusion of separately priced add-ons Linux servers are virtually a must in the delivery tier, unless you're not concerned about future scale-out Complex and potentially expensive licensing scheme Few consulting partners outside of the Nordics and Northern Europe Scalability of most recent version may not be heavily tested
Potential Fit	Community Oriented, Informational, Multichannel Publishing	Informational Sites, Basic Digital Marketing, Microsites & Landing Pages	Informational, Ultra-Large Single Sites	Multichannel Publishing, Ultra-Large Single Site
Unlikely Fit	Global Enterprise, Advanced Marketing Portal, Global/Enterprise Digital Workplace	Digital Workplace, Global Enterprise, Multichannel Publishing	Global/Enterprise Digital Workplace, Advanced Marketing Portal	SMB/Departmental or Global/Enterprise Digital Workplace, Advanced Marketing Portal
Compare To	Sitecore, Ektron, Hannon Hill, Microsoft, EPiServer	CrownPeak, Ektron, Escenic, Sitecore, CCI, OmniUpdate, Drupal	eZ Systems, Hippo, Adobe, e-Spirit	CoreMedia, Polopoly, Magnolia
Operating Systems	Windows and IIS in Management environment, can use Windows, Solaris, or Linux Delivery environment	N/A	Linux, Mac OS, UNIX, Windows	Ubuntu Server LTS, Red Hat Enterprise Linux (2.6), Debian GNU Linux, Fedora Core Linux, FreeBSD, Sun Solaris, Microsoft Windows 2003/2000 Server
Repository	File System — no database required	N/A	JCR (Apache Jackrabbit)	Database: Oracle, MySQL, SQL Server
Client	Firefox, Internet Explorer, and Safari	Browser: IE and Firefox	Any browser	Thick client (Java) that runs on Windows and Mac with limited, browser-based in-context editing UI
App Platform	COM and .NET; Optionally Java in delivery tier	Hosted — tag library and APIs for developers	Java	Java/J2EE: Oracle, BEA, IBM, Resin, and Tomcat

(Continued)	Ingeniux	Upland Software/Clickability	Magnolia	CCI (Escenic)
Licensing	Server-based option – Median deal size is about \$50K, including first-year support. Managed hosted option – \$1,800 to \$3,900 per month, depending on number of system users. Cartella collaboration system is \$18K for 30K users	Based on users, websites, and volume of visitor traffic. Annual subscription fee ranges from \$50K to more than \$200K	Dual: GPL, or commercial license starting at US\$12,000 per year per server	Complex, starting at €30,000 and going up steeply from there
Ownership	Privately held	Privately held	Privately held; 65 employees	Privately held

Category Summary: Mid-range Products

Phase / Attribute	CrownPeak	Ektron	Enonic	e-Spirit	eZ Publish
Technology					
Technical Administration & Security					
Threat Prevalence					
Authentication & Authorization					
System Reporting					
Multisite Management					
Cloud Services					
Development					
Configuration & Customization					
Integration & Extension					
Content Modeling					
Templating					
Performance					
Back-end Performance					
Site Caching & Delivery					
Content					
Contributor Experience					
Overall Usability					
UI Accessibility					
Contributing Content					
Authoring & Transformation					
Tagging & Taxonomy					
Content Reuse					
Media & Document Management					
Repository Services					
Content Lifecycle					
Workflow					
Globalization					
Archiving & Compliance					

Phase / Attribute	CrownPeak	Ektron	Enonic	e-Spirit	eZ Publish
Experience					
Publishing					
Standards Adherence					
Multichannel					
Mobile					
Digital Marketing					
Site & Campaign Analytics					
Testing & Optimization					
Segmentation & Personalization					
Social Media Integration					
Promotional Campaigns					
Community & UCG					
Workplace					
Collaboration & Networking					
Dashboard					
Ancillary					
Site Search					
Online Forms					
Module Ecosystem					
Vendor Intangibles					
Vendor Services					
Vendor Professional Services					
Channel Partner Services					
Support & Community					
Strategy & Roadmap					
Viability & Stability					

Key	
	Product does not provide this feature in any meaningful way.
	Product provides this feature, but is not as mature as its rivals
	Product provides this feature
	Product excels at this feature, <i>relative to other products in the same category</i>
	Product masters this feature, <i>relative to other products in the same category</i>

Phase / Attribute	Hannon Hill	Ingeniux	Upland	Magnolia	CCI
Technology					
Technical Administration & Security					
Threat Prevalence					
Authentication & Authorization					
System Reporting					
Multisite Management					
Cloud Services					
Development					
Configuration & Customization					
Integration & Extension					
Content Modeling					
Templating					
Performance					
Back-end Performance					
Site Caching & Delivery					
Content					
Contributor Experience					
Overall Usability					
UI Accessibility					
Contributing Content					
Authoring & Transformation					
Tagging & Taxonomy					
Content Reuse					
Digital Asset Management					
Repository Services					
Content Lifecycle					
Workflow					
Globalization					
Archiving & Compliance					

Phase / Attribute	Hannon Hill	Ingeniux	Upland	Magnolia	CCI
Experience					
Publishing					
Standards Adherence					
Multichannel					
Mobile					
Digital Marketing					
Site & Campaign Analytics					
Testing & Optimization					
Segmentation & Personalization					
Social Media Integration					
Promotional Campaigns					
Community & UCG					
Workplace					
Collaboration & Networking					
Dashboard					
Ancillary					
Site Search					
Online Forms					
Module Ecosystem					
Vendor Intangibles					
Vendor Services					
Vendor Professional Services					
Channel Partner Services					
Support & Community					
Strategy & Roadmap					
Viability & Stability					

	Key
	Product does not provide this feature in any meaningful way.
	Product provides this feature, but is not as mature as its rivals
	Product provides this feature
	Product excels at this feature, <i>relative to other products in the same category</i>
	Product masters this feature, <i>relative to other products in the same category</i>

Scenario Fits Summary: Mid-range Products

Phase / Attribute	CrownPeak	Ektron	Enonic	e-Spirit	eZ Publish	Hannon Hill	Ingeniux	Upland Software	Magnolia	CCI
Simpler Site										
Informational	◐	◐	○	◐	◐	◐	◐	◐	●	○
Microsites & Landing	◐	◐	◐	◐	○	◐	◐	◐	◐	◐
Mid-Range										
Basic Digital Marketing	◐	◐	◐	◐	◐	○	○	◐	◐	◐
Mobile Site	◐	◐	◐	◐	◐	◐	◐	◐	◐	◐
Community Oriented	○	◐	◐	○	◐	◐	○	◐	○	◐
Complex Site										
Advanced Marketing Portal	○	○	○	○	◐	○	○	◐	○	○
Global Enterprise	◐	○	○	◐	◐	○	○	○	○	○
Multichannel Publishing	○	○	◐	◐	◐	◐	◐	○	◐	◐
Ultra-Large Single	○	○	◐	◐	◐	○	○	○	◐	◐
Digital Workplace										
SMB/Departmental	◐	◐	◐	◐	◐	○	○	○	◐	○
Global/Enterprise	○	◐	○	◐	◐	○	○	○	○	○

Key	
○	Product does not provide this feature in any meaningful way.
◐	Product provides this feature, but is not as mature as its rivals
◐	Product provides this feature
◐	Product excels at this feature, <i>relative to other products in the same category</i>
●	Product masters this feature, <i>relative to other products in the same category</i>

CrownPeak: CrownPeak CMS

www.crownpeak.com

Vendor at a Glance

Specsheet	CrownPeak: CrownPeak CMS
Geography	North America and Europe
What's New	<ul style="list-style-type: none"> • Integration with the Demandbase targeting platform for personalization • Major overhaul of the user interface
Strengths	<ul style="list-style-type: none"> • Longest-standing SaaS-based CMS • Company is finding a niche in multisite management • Comparatively easy to roll out new sites, including microsites • Globalization features are supported through integrations and translation workflows • Boasts some major intranet deployments • Strong and consultative customer support • Developers feel comfortable in the .NET habitat
Weaknesses	<ul style="list-style-type: none"> • Default interfaces favor power users and may need “dumbing down” • UI is English only • Historical lack of web experience management tools (e.g., campaign management and social analytics), makes it less suited for online marketing use cases • Decoupled architecture tends to mitigate against more interactive, personalized scenarios • New C# API is generally only used by new customers thus far • Dearth of collaboration, mobile, and social services • Page-oriented system does not encourage content reuse • Underdeveloped community hinders customer cooperation • Corporate strategy and development has a tendency to meander • Future of the company is somewhat uncertain in light of aging venture investment that may push a premature “exit”
Potential Fit	Informational, Micro-sites & Landing, SMB/Departmental Digital Workplace
Unlikely Fit	Ultra-Large Single, Advanced Marketing Portal, Multichannel Publishing, Community Oriented
Compare To	Upland Software, Ektron, Sitecore, EPiServer
Operating Systems	N/A
Repository	N/A
Client	Any browser supporting Silverlight
App Platform	Hosted platform with .NET, SOAP/REST APIs

Specsheet	CrownPeak: CrownPeak CMS
Licensing	Based on the size of repository and the number of contributors, it starts at US\$72K annually, exclusive of implementation and support options
Ownership	Privately held, VC funded

Summary

CrownPeak provides a hosted web content management offering. Architecturally, it is a decoupled solution by default, with content published to a separate delivery environment. Some enterprises (particularly larger ones) prefer this separation, while others prefer a more coupled management-and-delivery platform from the likes of Upland Software (Clickability), or they prefer more traditional, on-premise software from Ektron or Sitecore.

In terms of capabilities and pricing, CrownPeak CMS rivals most mid-market players as well as its arch-competitor, Upland Software.

CrownPeak offers a sophisticated feature set and a limited development “playground” for customizing certain features yourself. However, it is certainly not cheap. It seems inexpensive in the beginning, especially if you’re looking at a single-domain license, and is attractive if you don’t want to invest capital in new technology and infrastructure; however, its total costs could rival traditional, on-premise software.

CrownPeak is suitable for a wide range of scenarios. It’s a decent candidate for Informational Sites, Microsites & Landing Pages, and to a degree Basic Digital Marketing sites, because it’s comparatively easy to roll out quickly. It is also a decent bet for SMB/Departmental Digital Workplace sites that don’t require substantial connectivity to internal applications, though it may be a stretch for something that is more global in scale or requires more than simple integration. We wouldn’t go so far as to recommend it for complex Global Enterprise, Advanced Marketing Portal, or Multichannel Publishing sites, because its lack of native XML management is a hindrance — particularly to Multichannel Publishing.

Note that with hosted offerings, vendor intangibles play a more important role when compared to traditional, on-premise software. The hosted model has advantages of time to market, subscription-based pricing, and the fact that you don’t need to own software, hardware, or IT resources. However, the disadvantages are that you lose the ability for advanced customization and extensions, and you introduce risks associated with outsourcing. Make sure to explore CrownPeak’s “Vendor Intangibles” on page 505 vigorously.

CrownPeak is venture funded through Alliance Bernstein, Sigma Partners, and Altos Ventures. It has secured two rounds of VC funding, and at some point is going come under pressure to sell itself to a larger firm so that those investors can “exit.”

Scenario Fits	
Simpler Site	
Informational	<input checked="" type="checkbox"/>
Microsites & Landing	<input checked="" type="checkbox"/>
Mid-Range	
Basic Digital Marketing	<input checked="" type="checkbox"/>
Mobile Site	<input checked="" type="checkbox"/>
Community Oriented	<input type="checkbox"/>
Complex Site	
Advanced Marketing Portal	<input type="checkbox"/>
Global Enterprise	<input checked="" type="checkbox"/>
Multichannel Publishing	<input type="checkbox"/>
Ultra-Large Single	<input type="checkbox"/>
Digital Workplace	
SMB/Departmental	<input checked="" type="checkbox"/>
Global/Enterprise	<input type="checkbox"/>

The company itself has a different plan. In 2012, CrownPeak expressed interest in fulfilling an Initial Public Offering (IPO) within 3 years. We will monitor this carefully and as a potential customer, you will want to as well. The company feels a bit small for an IPO, but the three-year deadline is probably quite real. In any case, changes will come. In the meantime, CrownPeak remains one of the few privately owned SaaS CMS vendors in light of the acquisition of Clickability in 2011.

Introduction

CrownPeak was built by former systems integrators who were frustrated with the kludgy character of CMS offerings at the time (circa 2000). It offers a sophisticated layout and many thoughtful choices for business managers and content contributors, alike. For example, the user inbox enables you to sort tasks according to various attributes (a handy benefit for busy editors and managers).

Initially, CrownPeak was completely focused on web content management, leaving experience management capabilities to integration options with a number of best-of-breed partnerships. These integration options tend to fit the needs of its traditional target, the upper mid-market tier. CrownPeak’s approach is to target enterprise marketers that need to get small sites up and running quickly to support simple marketing campaigns. In 2009, the company launched an optional “Landing Page Manager” — an offering similar to its main competitor, Upland Software.

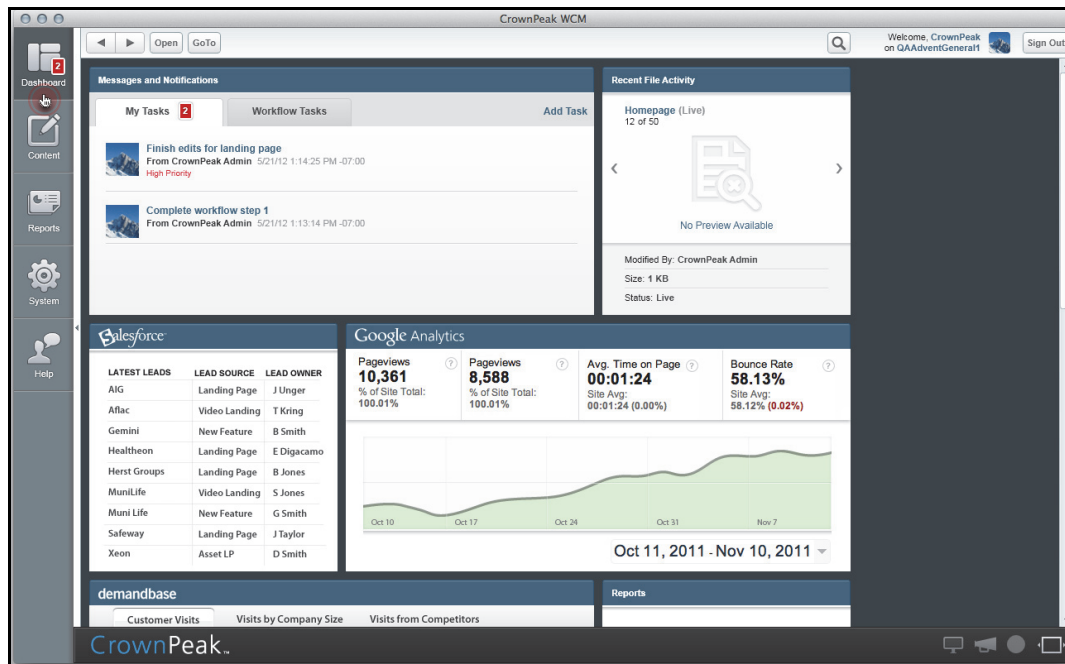


Figure 233. After about a decade of no significant changes to the interface, the UI that was launched in 2011 features a dashboard in CrownPeak CMS. Admittedly it’s cleaner and more modern now, but users say it takes time to readjust to new ways of finding and doing things.

After abandoning earlier initiatives to target lower-end customers, in 2011, CrownPeak launched a product called Web Content Optimizer (WCO). WCO is a standalone offering (but

can be integrated with WCM) to provide capabilities for targeting content, based on user attributes and behavior.

Technology

Technical Administration and Security

Content is managed on CrownPeak's servers, which then deploys rendered pages to your production sites or databases. CrownPeak offers several ways to accomplish this (we cover publishing in a later section). It is a page-oriented system with content exposed as folders and files. Ultimately, it is stored in a database, but because users are more familiar with a file/folder paradigm, it is exposed that way.

CrownPeak offers a hosting service as part of the standard licensing fee. The hosting service is provided exclusively in partnership with Amazon. Therefore, if you prefer other providers, you are out of luck.

CrownPeak now has five datacenters globally (for WCM and the content delivery infrastructure), up from a single datacenter in 2008. This should mitigate some concerns about the lack of datacenter coverage, but not necessarily redundancy. CrownPeak suggests that as well as utilizing Amazon EC2 for some of this capacity. In addition, there is another external fail-over facility (developed in-house) that CrownPeak claims will protect you from the sort of outages that have affected Amazon's service of late. Our advice: be sure to test how redundancy and failover work in your scenarios.

The publishing capabilities are powerful; you can publish to another server statically or dynamically, via FTP, HTTP and SOAP/REST Web Services or to a database via ODBC. There's also a publishing monitor, which shows details of different publishing processes (along with multiple publishing reports).

Like other publishing-oriented systems, resolving many dependencies and relationships can become complicated: a seemingly simple update can affect other content in ways that aren't always easy to foresee. Customers claim that publishing in CrownPeak works in "mysterious" ways. Spend time exploring these aspects, particularly incremental publishing, publishing performance (and its impact on editing activities), and behavior when content is updated or deleted.

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

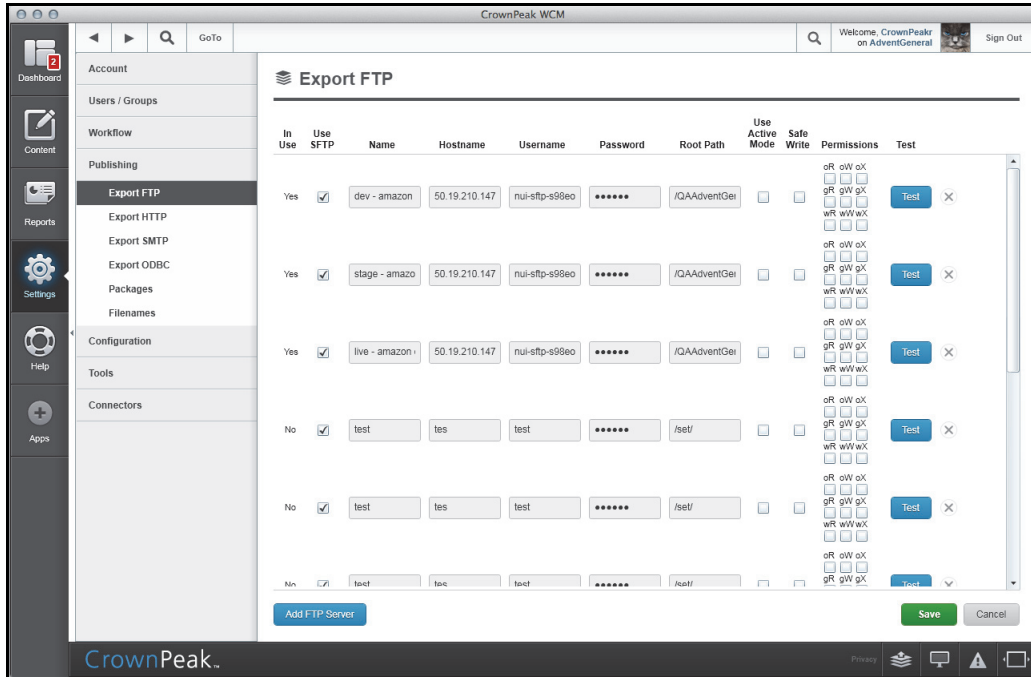


Figure 234. You can create publishing destinations and set up servers via the admin interface.

Role management is quite granular, and CrownPeak includes some very practical touches, such as the ability to circumscribe logins to certain IP addresses, or even to specific times of the day. Of course, as with any hosted solution, you may face various challenges if you want to leverage your preexisting directory systems, such as LDAP or Active Directory.

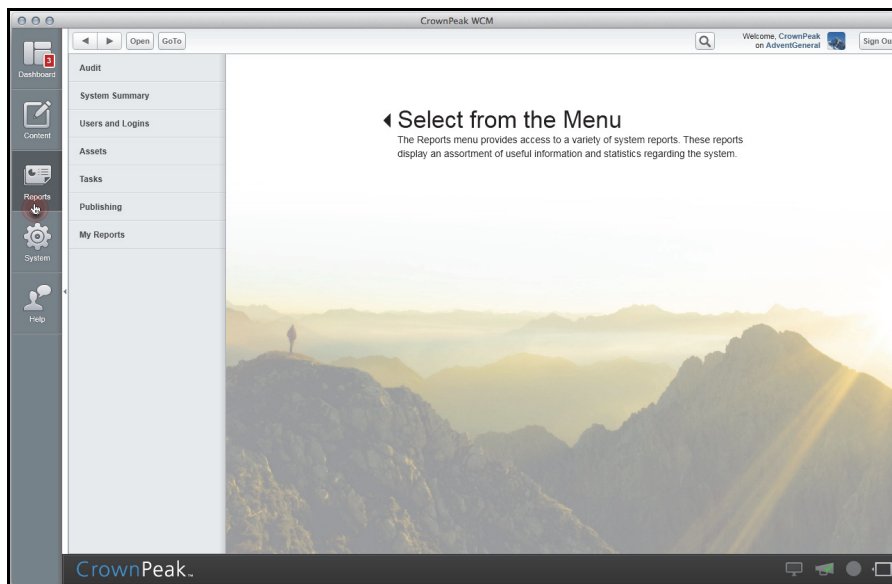


Figure 235. A small variety of system reports is available to CrownPeak administrators in the CMS UI.

A key strength of the product is its extensive reporting features. Business managers can report on a host of metrics, including workflow clearance (Figure 236), document and asset check-outs, login patterns, and so forth. One report we liked was “Never Logged In.”

Type	Label	Status	Changed By	Changed On	Size	Template	ID
	401(K) Plans	Live	CrownPeak Admin	7/3/12 3:40:15 PM	2 KB	Content Page	2316
	Analysis and Structure	Rejected	Fahd Shaaban	5/18/12 3:40:00 PM	2 KB	Content Page	2367
	Analysis Structure	Stage	Mark Christiany	8/8/12 1:40:23 PM	2 KB	Content Page	2326
	Annuities	Stage	CrownPeak Admin	7/3/12 3:37:58 PM	2 KB	Content Page	2319
	Article Page	Draft	Tim Griffiths	6/13/12 6:22:05 PM	3 KB	Article Page 1-2...	2559
	Article Page Column 1	Stage	Fahd Shaaban	5/22/12 12:20:34 PM	3 KB	Article Page 1...	2525
	Article Page Column 2	Stage	CrownPeak Admin	3/21/12 1:37:52 PM	6 KB	Article Page 2...	2523
	Article with widgets	Stage	Tim Griffiths	7/3/12 2:51:33 AM	3 KB	Article Page with...	2961
	Asset Allocation	Stage	Jason Yoo	4/3/12 5:13:45 PM	1 KB	Content Page	2262
	Claims Settlement	Stage	Jason Yoo	4/3/12 5:13:46 PM	2 KB	Content Page	2329
	CLONE	Draft	CrownPeak Admin	12/1/11 3:54:58 PM	2 KB	Content Page	2366
	Company	Live	Jason Yoo	5/4/12 3:34:32 PM	2 KB	Content Page	2289
	Consulting	Stage	Jason Yoo	4/3/12 5:13:46 PM	2 KB	Content Page	2264
	Copart Example	Draft	CrownPeak Admin	5/25/12 11:31:04 AM	530 KB	WYSIWYG	3946
	Customer Testimonies	Stage	Jason Yoo	4/3/12 5:13:46 PM	3 KB	Content Page	2191
	DemandBase	Live	Fahd Shaaban	8/14/12 9:43:25 AM	247 B	Article Page 1-2...	3054
	DemandBase Form	Live	Fahd Shaaban	7/23/12 1:20:34 PM	8 KB	WYSIWYG	4064
	Financial Consulting	Stage	Jason Yoo	4/3/12 5:13:46 PM	2 KB	Content Page	2330

Figure 236. A Content report, which includes workflow clearance.

Development

Unlike most other SaaS-based tools, CrownPeak gives you broader access to a controlled sandbox. This was traditionally done via VBScript, but the company has started offering a more modern C# API. The .NET API was released in early 2011, and has yet to garner significant levels of adoption. According to CrownPeak, only new customers are currently using the C# API. If you’re one of the older CrownPeak customers, you are probably still running on COM. While it is not uncommon to see major overhauls of APIs among CMS vendors every now and then, quite often this transition brings its fair share of bugginess; test it carefully.

Developers use this API to create templates. A template in CrownPeak lingo is a container for one or more template files. Each template file consists of HTML, combined with API calls that implement some kind of functionality. For example, a template file “input.asp” defines the form for capturing content via a form, while a template file “output.asp” defines how that content will be displayed.

The templating construct is a bit awkward at first, but you can use it to publish effectively to multiple sites and devices. Specifically, you can assign multiple “layouts” to the same content, e.g., intranet/extranet/mobile site, etc. The templates “find” content according to business logic, rather than the content when finding templates.

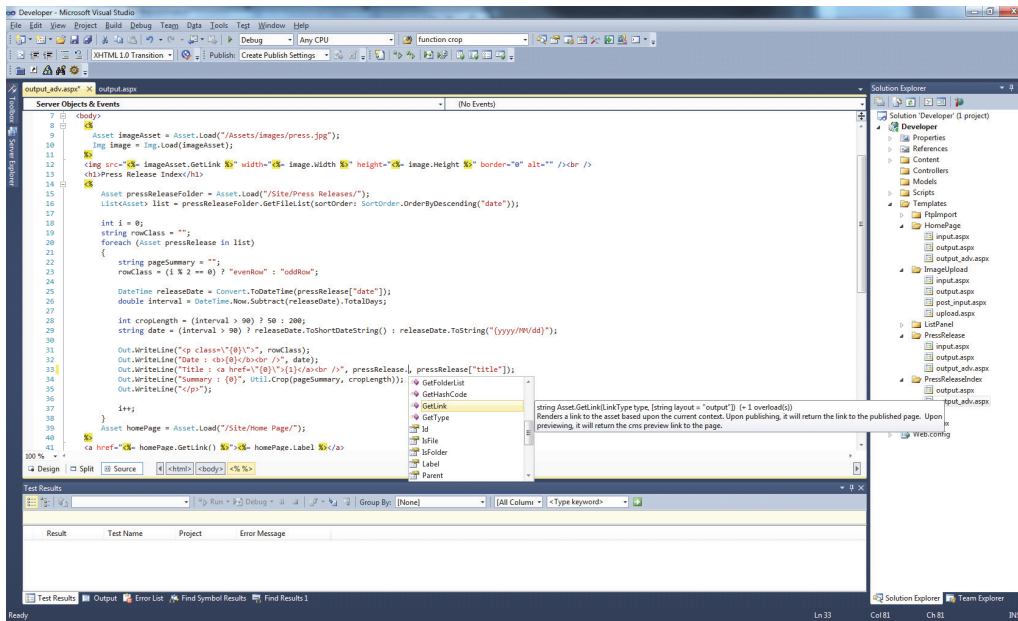


Figure 237. You can create templates using the web interface, or use an IDE such as Visual Studio or Eclipse.

Developers use a browser to create and edit these templates. While most functionality is available across several browsers, the experience varies; test particularly well if you use a Mac browser. In response to developers that did not like to work in CrownPeak’s original, browser-based IDE, CrownPeak offers its “CrownPeak Desktop Connection” feature, and it is much appreciated by developers working with this CMS. After installing this application, the Desktop Connection enables template development using familiar tools: editors like Dreamweaver and TextPad, as well as more robust tools like Visual Studio and Eclipse. Developers then upload them to the CrownPeak CMS through a Web Services connection.

Developers have access to a set of pre-defined code elements and templates that they can reuse in their “Code Wizard.” Unfortunately, “Code Wizard” still runs on the old VBScript API, and none of the code libraries can run server-side due to the decoupled nature of CrownPeak CMS.

The product also comes with a .NET integration kit to integrate with internal or third-party systems. CrownPeak also offers integration hooks into several other major hosted software vendors, such as salesforce.com (Sales Force Automation), Oracle/RightNow Technologies (CRM), and ExactTarget (email broadcasts). Recently added is integration with Demandbase’s personalized targeting platform, which provides API-level access to a service that enriches visitor information by ways of reverse IP look-up and company segmentation information. This can then be used to assist in targeting dynamic content to certain market segments, such as knowing what vertical market a page visitor came from can be used to influence delivered content. You also can access CrownPeak functionality via REST APIs. This is very useful in integration scenarios, such as exposing CrownPeak content into an external application like salesforce.com.

The learning curve is just steep enough that a non-technical webmaster may not be able to customize the templates. However, most other configurations are accessible through a

browser. Like other hosted players, CrownPeak handles customizations for most of its customers.

Performance

There are two environments are particularly affected by performance: the live website and the WCM environment. CrownPeak is responsible for WCM environment, and the performance of the live site is up to you (unless you have outsourced that to CrownPeak).

CrownPeak offers SLAs, monitors, and it guarantees performance. However, one customer we talked to mentioned poor performance during publishing, as well as occasional issues in the editorial interface. To be fair, slow “baking” times frequently affect on-premise tools that feature decoupled architectures. On the positive side, customers praised CrownPeak for proactively monitoring and identifying performance issues. Make sure to test out this aspect of CrownPeak — especially if you have a globally distributed team.

Content

Contributor Experience

The CrownPeak UI has been described as “very pedestrian,” albeit simple enough to get the job done. Some customers have reported that this interface is intuitive enough to make it easy to train content producers of varying technical skill levels, including non-English speaking users. However, this can also become confusing, especially for non-power users. It will take time to get to know which options display under what menu. It is also sometimes confusing and not very clear why a particular option displays where it does.

In 2012, CrownPeak began to roll out a major overhaul of the CMS user interface, which had shown its age in lieu of its lack of significant updates in about a decade. Like many other vendors, the UI is based on the notion of a personalized workspace; it can be set up on a per-user, per-role basis to reflect typical tasks and content types, for example.

There is a set of modules that can be snapped into the UI dashboard, and these modules can differ based on specific roles within your organization. They include integrations with Google Analytics, Salesforce, and the aforementioned integration with Demandbase. The modules can be controlled via AD/LDAP profiles.

Help information (including videos) is available within this interface, with walk-throughs on common tasks for new users. Internationalization is still not particularly strong since CrownPeak still caters to its predominantly US/UK customer base; the interface is only available in English.

CrownPeak promises that these recent improvements are just the beginning and a raft of other enhancements should be part of the UI in 2013, including an improved notification system and “right-click” mouse actions for common shortcuts. However, roadmap promises have no

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

guarantees; use caution before putting too much store in them, particularly around delivery dates. It’s always more sensible to buy software based on what actually exists in the product — not what’s demonstrated to you in PowerPoint presentations.

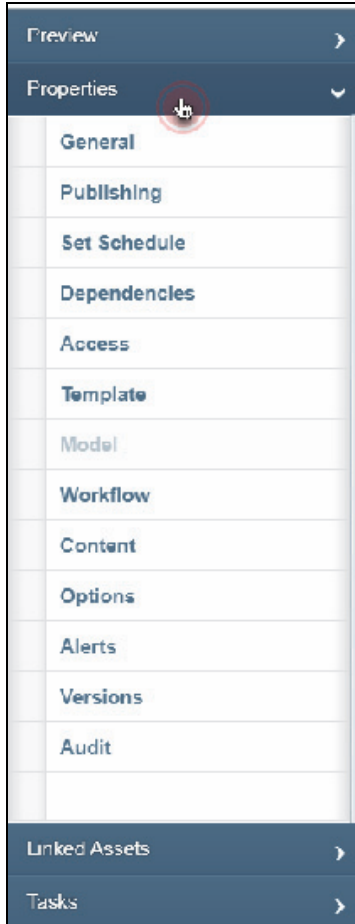


Figure 238. The Properties tab is an example of where users could get lost in the UI; it contains everything from template properties to alerts.

Note that the interface is based on Silverlight, which means you need Silverlight plugins for all contributors. Only a handful of other vendors we cover (and none in the CMS space — including Microsoft) has committed to Silverlight as a user interface layer.

Contributing Content

For authoring, CrownPeak supports either a third-party, applet-based editor from Ephox, or TinyMCE. Ephox is the same editor that is OEM’d in other CMS offerings and is well regarded. However, Ephox EditLive! is a Java applet that must be installed (this is a switch from CrownPeak’s previous emphasis on ActiveX controls, which provided a rich application interface). Alternatively, you could switch to TinyMCE as the editor if you have problems supporting Ephox’s Java requirements. In either case, some fine-tuning for optimal editing results (e.g., Microsoft Word cut/paste cleanup) will be necessary.

By default, CrownPeak is a page-based system. When you create content, you are creating a page and must be aware of how the page looks and how navigation and header images display. Despite the availability of so-called “snippets,” the page-based architecture makes it difficult to separate content from presentation, and repurposing can be a big challenge — especially when you are creating content for multiple destinations with different designs and layouts (think web vs. mobile, for example). Similarly, it is not easy to track where content is used in other sites or pages via a “where used” report, which would show how your changes would affect content and where. As one of the workarounds for this, CrownPeak suggests using the authoring interface toggles between

different content views to see exactly where a content element is used in other pages. Note that this is a functionality you would have to create yourself in the UI or in your staging environment.

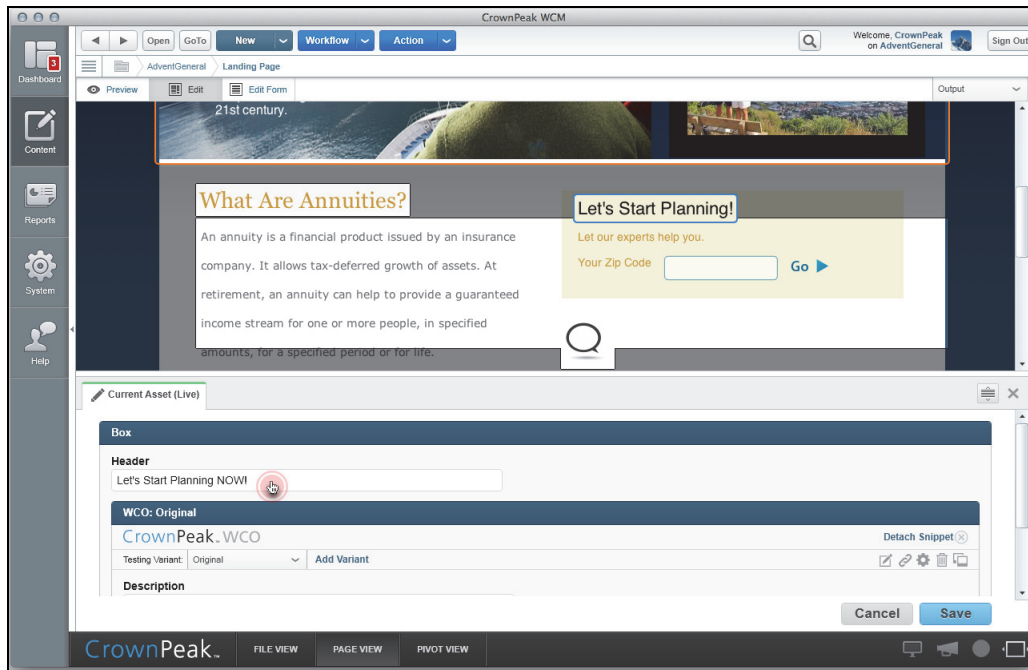


Figure 239. CrownPeak's inline editing experience: The default WYSIWYG editor is a Java applet and provides most features, although it's spotty on a Mac.

You can edit content in-context (but only within the editorial environment and not the live environment). Depending on your requirements, you can decide to open the content in a full-fledged, forms-based interface, or use inline editing for quick editing. However, to enable these features, you must include the appropriate code in your templates. CrownPeak CMS stores internal hyperlinks as unique objects, so the system can provide link management. When content deploys, CrownPeak scans for dependency and updates link references. Note that this can drag publishing time. Authors can also contribute content via email messages that contain special tokens, and then these messages are parsed and read into a specialized template.

Like HP, CrownPeak provides branching. Branching essentially copies a page and ties it to the original version. This is a useful feature if you want to work on different versions of a page concurrently, without waiting for one version to go live. The same principle applies to creating language variants or variants that will be used for personalization or multi-variant testing.

You will look in vain for any page-builder functionality. Authors are locked into editorial forms that are tightly bound to developer-mediated templates. Some companies may prefer this control and predictability, while other web publishing operations need systems that enable business users more control over ad hoc page development.

The screenshot displays the 'Form Builder' interface. It features a list of form fields with the following details:

- Field 1:** Field Name: First Name (Required)
Field Value: fname
Field Type: Text Box
- Field 2:** Field Name: Last Name (Required)
Field Value: lname
Field Type: Text Box
- Field 3:** Field Name: Email (Required)
Field Value: email
Field Type: Text Box (with a dropdown menu open showing options: Text Box, Email Text Box, Text Area, Drop Down, Check Box, Blank Area)

Below the fields is a 'Hidden Values' section with two input fields labeled 'field name' and 'field value'. At the bottom is a 'Right Side Nav' section with a 'Title' input field and a 'Title Url' field with 'Browse' and 'Clear' buttons.

Figure 240. CrownPeak's Form Builder.

CrownPeak has a reasonably sophisticated approach to handling taxonomies and metadata. You manage your classification system as a content type, with nodes represented as folders in the system that can be moved around, taking children with them. Therefore, changes can be versioned and put through a workflow. Additionally, the product supports multifaceted taxonomies. The author can classify content objects by assigning values from the “industry” facet. Note that the hierarchical screen is expandable, and that a searched term will auto-complete after typing the first few letters.

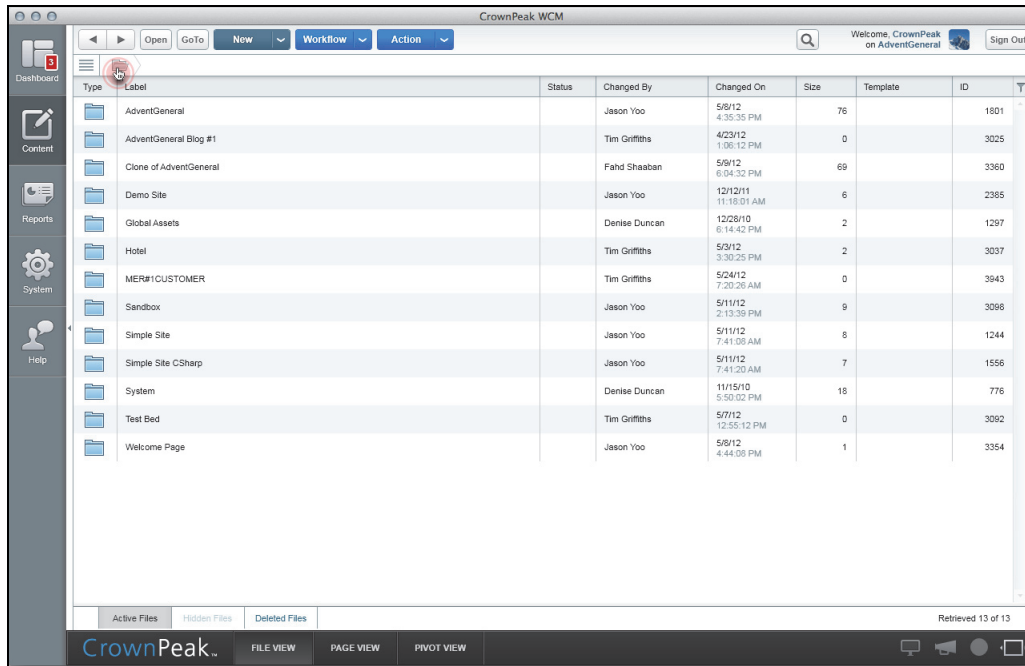


Figure 241. Folder structure view in CrownPeak CMS.

Digital Asset Management is one of the weakest areas in CrownPeak. While it is possible to store and version images as content items in the system, some unexpected problems can arise. For example, you can only resize assets when they're used in piece of content.

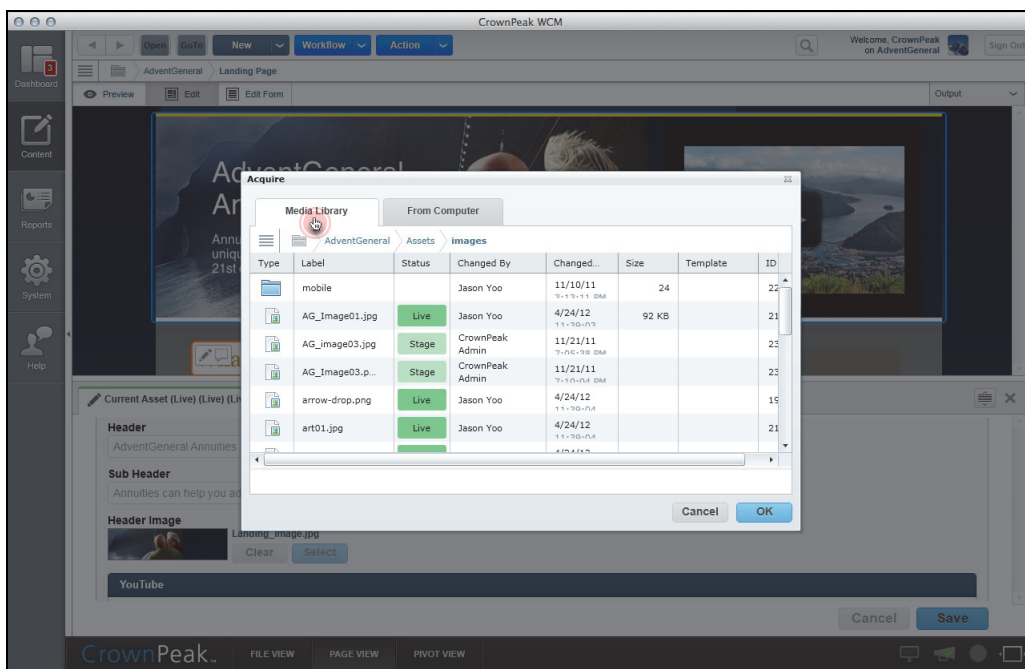


Figure 242. A view into CrownPeak's Media Library. Some users find that in terms of asset management, simpler tools like WordPress provide more functionality, in a more user-friendly way.

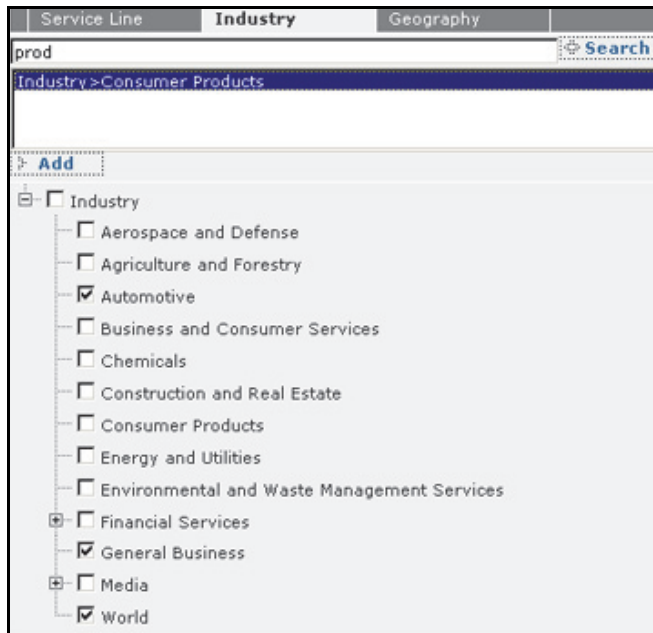


Figure 245. Classification sub-system.

database queries, and not proper search indexes; they may not pick up your Office files, for example.

Content Lifecycle

CrownPeak’s workflow provides sophisticated notification and an intelligent control panel, all with a forms-based workflow-modeling tool for administrators. Customers like the “conversational” nature of the workflow; user avatars are part of the interface, and you visually “see” whom you’re handing off to in the workflow. Additionally, color codes for different workflow states can enhance usability. However, the platform has limited ad hoc routing, which seems like a glaring omission in web publishing environments that favor flexibility over standardization.

Like Percussion’s CM content management system, CrownPeak uses a “state-transition” model, which is technically elegant and particularly well suited for highly collaborative editorial processes. However, it is not always intuitive to model and use.

Unlike Upland Software, CrownPeak targets customers with multisite management needs, including globalization scenarios. While native functionality for globalization is sparse, CrownPeak is known for integrations with third-party translation management systems.

CrownPeak’s “portfolio” model enables you to clone entire websites easily, including all settings and configurations. Then you can establish arbitrary inheritance between parent and child objects. CrownPeak’s interfaces for managing this seem cruder and more technical than some more advanced, on-premise tools.

On the down side, although authors can choose where content is published to, since it’s a very page-oriented system, you likely would not want to use CrownPeak if you create and manage highly granular, componentized content.

This becomes tricky for developers that need to figure out all possible sizes before images are uploaded into the CMS and added to site pages. Sizing dimensions have to be specified in the page template, and developers claim they cannot easily add other dimensions, since it requires going back to the template level and adding data templates for the new sizes that are needed.

On the plus side, repository services are reasonably strong, with effective versioning and version control. Customers report that repository search is user friendly, although you should note that these are simple

Experience

Publishing

Like most “baked” architectures, CrownPeak sites typically have friendly URLs, although ultimately that’s determined by your delivery system. It is also possible to set up and manage vanity URLs via programmatic access to URL pathing. Given the decoupled architecture, your developers may need to work out some additional magic if you require 301 redirects without the risk of losing inbound traffic.

In terms of mobile delivery, be prepared for more manual work and coding. CrownPeak doesn’t utilize any of the common mobile device detection libraries like WURFL. Instead, it employs user agents that route a limited set of predefined devices to different pages, which doesn’t mean that there are mobile-specific templates. Mobile device emulation to preview what your content will look like is only available for about 20 mobile screens.

Digital Marketing

To counter capabilities from Upland Software, CrownPeak beefed up its ability to extend traffic reporting to content managers, principally through integration with Webtrends’ hosted “Live” service. Webtrends reports can be integrated with existing CrownPeak reports, and Webtrends’ “SmartView” IE plugin can overlay traffic stats on content items, as it does with SDL. Notably, you also can perform content lifecycle management or promotional offers based on live traffic data, e.g., move a content item to a different part of the site after its hundredth view, or archive all items that have had no traffic.

CrownPeak also has integrated with Adobe’s (formerly Omniture) Test&Target. You can create content and publish directly out of CrownPeak, signifying with special buttons which fields and regions you want to test and target. Targeting rules must be set up in an interface other than – Adobe’s.

For simpler needs in web analytics, CrownPeak offers its own CrownPeak Analytics tool. While basic, the interface is relatively slick. It includes a feature called “MagicSpy,” which enables site managers to see what users are doing on the site in real time. More advanced features like click-path analysis and conversion rate views, however, are unavailable.

In terms of personalization and associated analytics, CrownPeak offers an additional layer of visitor reporting via integration with Demandbase. This is designed to enable you to gain additional insights into your web visitor base, such as which organizations they are from, and in which market sector those organizations operate. Ultimately, this allows you to use this analytical data to create groups and target content specifically toward those groups.

Experience	
Publishing	
Standards Adherence	<input type="checkbox"/>
Multichannel	<input type="checkbox"/>
Mobile	<input type="checkbox"/>
Digital Marketing	
Site & Campaign Analytics	<input type="checkbox"/>
Testing & Optimization	<input type="checkbox"/>
Segmentation & Personalization	<input type="checkbox"/>
Social Media Integration	<input type="checkbox"/>
Promotional Campaigns	<input type="checkbox"/>
Community & UCG	<input type="checkbox"/>
Workplace	
Collaboration & Networking	<input type="checkbox"/>
Dashboard	<input type="checkbox"/>
Ancillary	
Site Search	<input type="checkbox"/>
Online Forms	<input type="checkbox"/>
Module Ecosystem	<input type="checkbox"/>

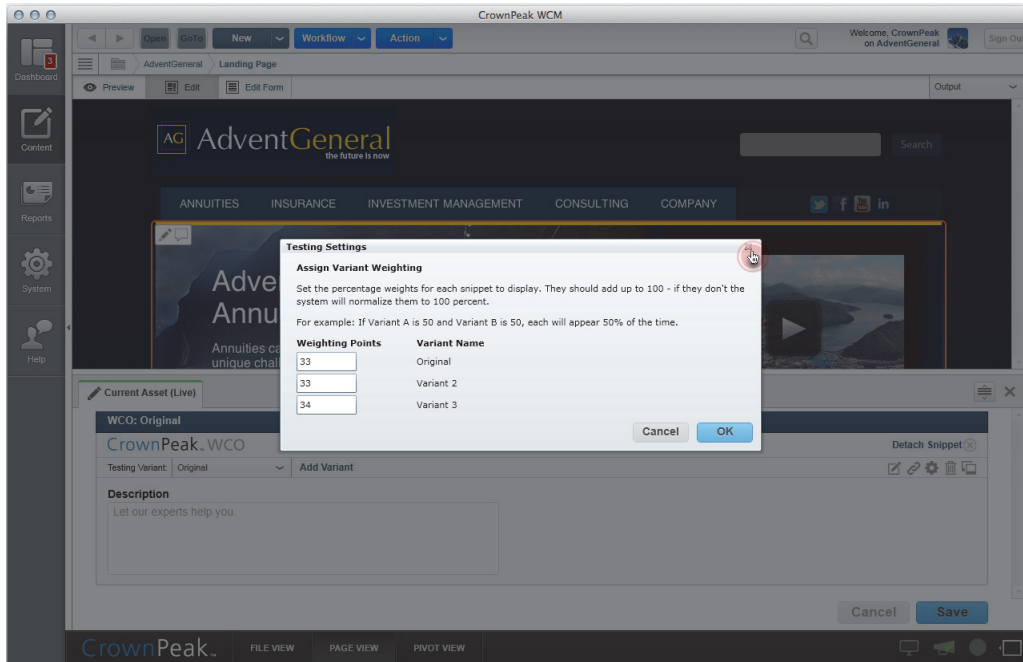


Figure 246. Assigning testing values in CrownPeak WCO.

Overall in the web experience management game, CrownPeak is significantly lagging behind the industry in terms of providing functionality for UGC, blogging, commenting, social media, and multichannel delivery (e.g., mobile). Blogs, comments, polls, and surveys can be addressed via template manipulation — a comparatively more involved process within the industry that has long since moved to widgets and pre-built integrations. To be fair, there are some things you can do; you can configure your website comments to be ingested into the CrownPeak repository via SMTP, where you can apply workflows and re-publish them. If you want to auto-tweet in certain cases (e.g., a press release), you need to specify that action in the output template with a call to the Twitter API. Inconveniently, the call will be executed every time you publish a page.

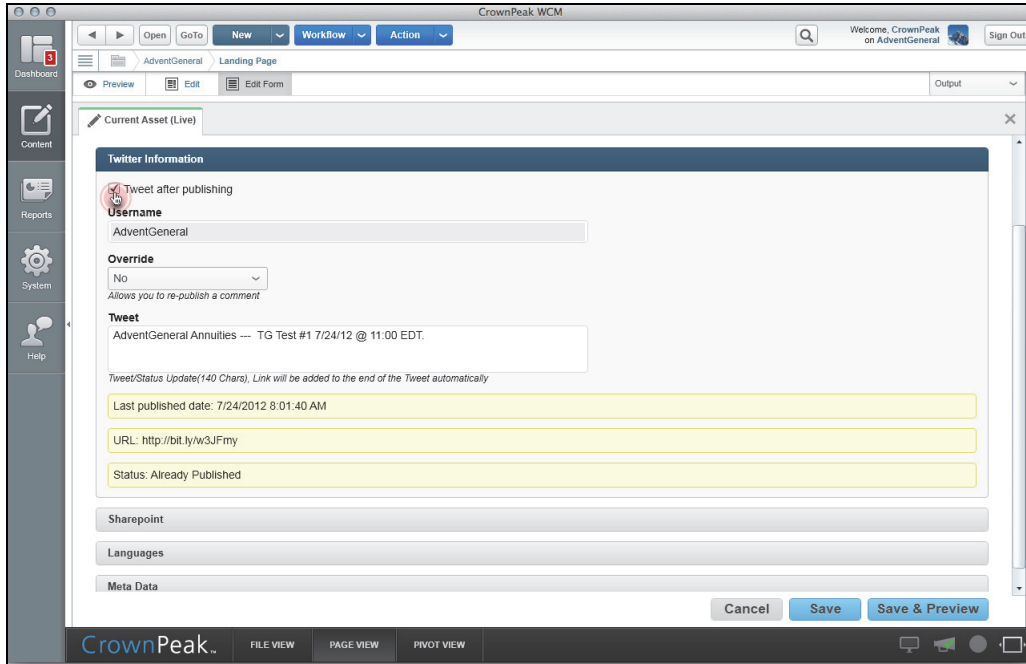


Figure 247. Rudimentary Twitter publishing capabilities in CrownPeak CMS.

CrownPeak also offers functionality to treat Twitter and Facebook status messages as managed content, enabling authorized staff to publish to enterprise social networks — including via a workflow — with versioning and archiving. In general terms, the majority of actions you take with social networking services like Facebook and Twitter will require all-manual data entry, including the URLs.

Vendor Intangibles

CrownPeak serves its customers mainly from North America and the UK, and that’s where most of its approximately 50 employees are located. There’s also some presence in Tokyo, but there is no “official” physical office. CrownPeak does not share financial information publicly, but says that they are increasingly targeting companies with revenues greater than \$1 billion. According to the company, 80 percent of its revenue comes from service licenses, and twenty percent is from professional services work.

CrownPeak uses a mix of in-house professional services and partners for implementation services. They have a dedicated “Web Consulting Services” group (numbering around 16 people) that provides implementation and consulting services. Currently, about 50 percent of its implementations are in conjunction with integrators and digital agencies.

CrownPeak has a formal community network (connect.crownpeak.com). This network contains code snippets and API changes, as well as tips and how-to videos.

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

With a hosted service provider, the non-functional aspects become increasingly important than they would be for traditional in-premise software vendors; CrownPeak is no exception. In a hosted model, the vendor controls everything. You are also vulnerable if the vendor goes out of business. Even if you have an XML backup of your content and maybe even the live site, that's different from having a working WCM environment. In fairness, these are consistent issues for all hosted players.

Customers have given high marks to CrownPeak for customer support (which runs 24/7/365 as of Q4 2012), proactive performance monitoring, and for resolving issues. Customers seem unusually happy with the level of support that they have, and are impressed with how quickly new sites can be deployed on the platform. CrownPeak is reaching out to partners in an effort to scale implementation services. While potential partners must go through a certification process, it remains to be seen if they can sustain CrownPeak's notable level of support.

The cost varies depending on the numbers of users, sites, and pages. Licensing flavors vary from one domain to enterprise levels. The basic pricing starts around \$72K per year, exclusive of extra modules, support, implementation, and training. Under this basic license, you can have up to 1000 published pages, 5 users and up to 20 microsites under one domain name. The CrownPeak Enterprise license covers an unlimited number of domains, up to 500 users, and up to 250,000 pages. The price can vary in the neighborhood of \$200K.

You can buy lifetime active support for around \$20K that enables you to have a set number of support hours per month throughout the life of your setup.

Implementation services costs range from three to five times the license cost, depending on the complexity of the implementation and how much do-it-yourself templating you're willing to endure.

CrownPeak is venture funded through Alliance Bernstein, Sigma Partners, and Altos Ventures. It has secured two rounds of VC funding, and may soon feel some pressure to sell itself to a larger firm so that investors can "exit." CrownPeak remains one of the very few privately owned SaaS CMS vendors. CrownPeak too may soon find a new corporate home, either via an acquisition or (as the company itself seeks) via a planned IPO that is slated to happen in the next three years. With any of these claims, you have to be extra vigilant; CrownPeak has not been exhibiting the level of growth necessary to lend itself well to a successful IPO. Secondly, any change is a disruption, which may cause multiple changes for present and future clients.

Conclusion

Over the past decade, CrownPeak has proven that it can support satisfied customers in the upper mid-market tier. Just note that part of the value of what you're buying with this service is their consultative approach. You'll likely find the company's well-regarded support more appealing than the snazzy (but generally underutilized) API.

Historically, CrownPeak has not built web experience management tools as part of its content management offering. Rather, they have opted to partner, rather than to build. Now they offer either some very basic baked-in tools, or their traditional model of integrating with best-of-breed partners. If you go with the best-of-breed partner method, you may need to manage multiple relationships with different hosted suppliers (with the possible benefit of obtaining better tools and services as a result).

In general, CrownPeak customers seem satisfied, but prospective licensees may be underwhelmed — especially at this price point. You'll want to balance the traditionally strong

customer service found in this sort of SaaS offering against a product that often lags behind its competitors in terms of functionality. CrownPeak remains a comparatively small vendor with outsized ambitions, but it has only a modest track record of development success. Consider CrownPeak first and foremost if time to market is more important than rich features and long-term TCO.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Ektron: Ektron CMS

www.ektron.com

Vendor at a Glance

Specsheet	Ektron: Ektron CMS v8.7
Geography	Global
What's New	<ul style="list-style-type: none"> • Aloha as default rich text editor in lieu of Ektron's own eWebEdit400 Editor • Cloud deployment options on Windows Azure and Amazon EC2 • Enhanced SharePoint integration
Strengths	<ul style="list-style-type: none"> • Broad range of out-of-the-box features, with special emphasis on fast-evolving social tooling • Support for reusable widgets (.NET controls) gives web designers flexibility in creating complex pages quickly • eSync module enables content deployment across environments, round-trip synchronization, and other related deployment tasks • Unlike Sitecore and EPiServer, company sells products and consulting directly to customers • Large, active developer community • Customers report good first-line product support
Weaknesses	<ul style="list-style-type: none"> • Ektron tends to do things its own way, rather than via industry norms • Comparatively limited and not user-friendly multilingual capabilities • LDAP integration lacks depth, Single Sign-On (SSO) capabilities are not available • Setup, deployment, configuration, and templating can be complicated; architecture is older and less standardized than that of some competitors • Emphasis on packaged features comes at the expense of developer extensibility, relative to its direct competition • Complaints of channel cannibalization have led some consulting partners to re-assess their involvement with Ektron • Ektron has a history of releasing under-tested code in its rush to add more features
Potential Fit	Basic Digital Marketing, Microsites & Landing Pages, Community Oriented, SMB/Departmental Digital Workplace
Unlikely Fit	Advanced Marketing Portal, Multichannel Publishing, Global Enterprise, Ultra-Large Single Site
Compare To	Sitecore, Kentico, EPiServer, Telerik, SDL
Operating Systems	Windows
Repository	Database: Microsoft SQL Server
Client	Browser: IE, Chrome, Safari, and Firefox on Win, Mac, and Linux
App Platform	.NET

Specsheet	Ektron: Ektron CMS v8.7
Licensing	Combination of URLs and contributors. 1 URL/5 users edition starts at US\$15K, Enterprise Edition \$100K and up. Add-on modules sold at extra 20% of the base license fee each.
Ownership	Privately held

Summary

Nashua, New Hampshire-based Ektron started off in 1998 as a small upstart in the Web CMS business, barely noticeable against industry background noise. However, Ektron emerged from challenger status to become a serious contender in the middle-tier Web CMS market, filling much of the void left by Microsoft’s storied indifference to web content management technology.

Ektron wasn’t an “overnight” success; it took years to get to the point of having a solid offering at an attractive price point (with a feature set driven largely by customer feedback), and that’s what Ektron has produced in the CMS400.NET product. In late 2011 – early 2012, Ektron started to eradicate this meaningless product name (perhaps only meaningful to the New Hampshire crew) in favor of just calling it Ektron CMS. However, you can still see “CMS400.NET” on its website, in the UI, and in some documentation. Old habits die hard.

However, what Ektron has not delivered is an integrator-friendly offering, like Sitecore or EPiServer. Even though the company has a handful of implementation partners, Ektron increasingly completes projects itself in its traditional North American bastion.

Ektron’s .NET-based CMS is well suited for simple to mid-range websites such as Basic Digital Marketing sites, Microsites, and Interactive Marketing sites. It can be pressed into other roles (with time and effort), and made reasonably scalable (again, with time and effort), but we believe the sweet spot tends to be in mid-range scenarios.

Slightly ahead of many of its peers in the same market segment, Ektron focused early on community-oriented and intranet sites that made heavy use of wikis, forums, blogs, tagging, polls, ratings, and social features. Take note, however, that this is a product in which delivery and management are by nature coupled; the same software powers internally and externally facing features. That’s an unpopular architecture for some customers.

Customers tell us that Ektron is less suited as a platform for custom development, although the product continues to improve in this respect. Ektron has somewhat belatedly expanded its APIs and Visual Studio integration, but in general the platform is more “out of the box” — if less standardized — than Sitecore or SDL.

Scenario Fits	
Simpler Site	
Informational	<input checked="" type="checkbox"/>
Microsites & Landing	<input checked="" type="checkbox"/>
Mid-Range	
Basic Digital Marketing	<input checked="" type="checkbox"/>
Mobile Site	<input checked="" type="checkbox"/>
Community Oriented	<input checked="" type="checkbox"/>
Complex Site	
Advanced Marketing Portal	<input type="checkbox"/>
Global Enterprise	<input type="checkbox"/>
Multichannel Publishing	<input type="checkbox"/>
Ultra-Large Single	<input type="checkbox"/>
Digital Workplace	
SMB/Departmental	<input checked="" type="checkbox"/>
Global/Enterprise	<input checked="" type="checkbox"/>

In 2011, Ektron decided to overhaul its own proprietary developer APIs completely into the so-called Framework API to support traditional CRUD operations. While the new API seems straightforward, developers working with Ektron may have to spend some time to adjust to it.

Some customers will welcome Ektron’s emphasis on productized features over extensibility. For others, it will present a drawback. It’s something to investigate closely against your extensibility requirements.

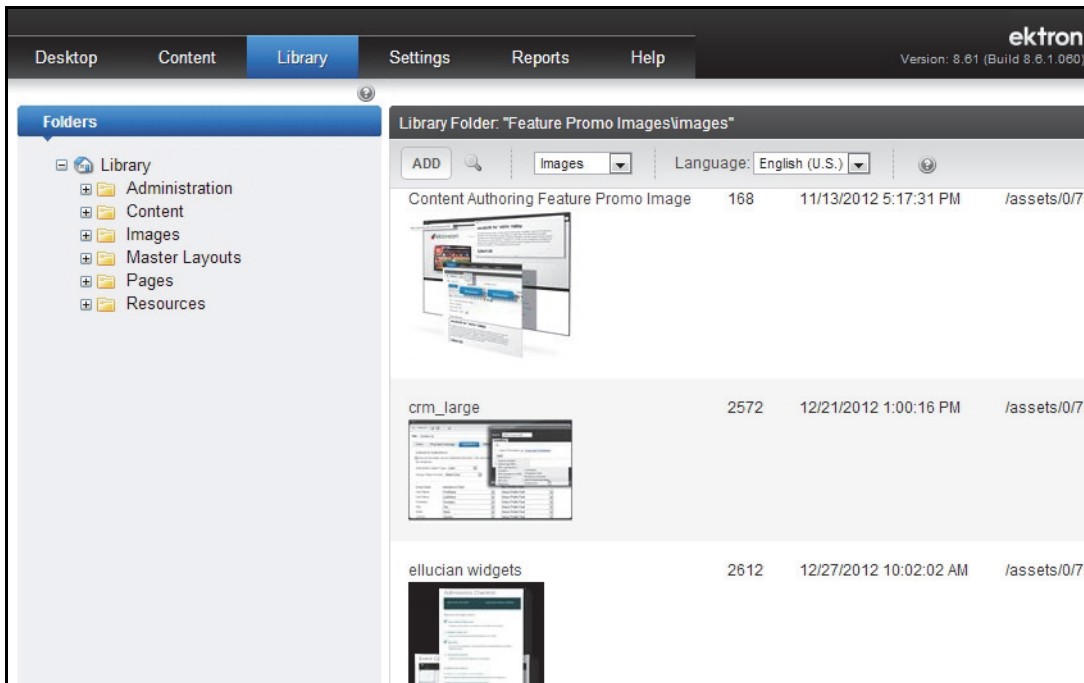


Figure 248. Ektron’s UI.

Another area you should investigate is Ektron’s multilingual capabilities, if your website is presented in more than one language — more than in English, to be precise. From the early days, as well as given their historic roots in the U.S., Ektron has not put much emphasis on multilingual content management. If localization and translation are not foreign to your requirements, keep in mind that Ektron’s UI only comes in English and French. While you can clone a page authored in U.S. English and create a Dutch version of it (for example), this experience is not as easy or efficient as it is in some of Ektron’s competition.

Introduction

Founded in 1998, Ektron was known initially for its “eWebEditPro” editor, one of the first browser-based WYSIWYG “rich text” editor controls. Ironically, this ActiveX-based HTML editor — initially the company’s only product — gave Ektron a huge presence in the CMS world via OEM relationships with other vendors. It wasn’t long before Ektron saw the potential in developing its own CMS, and in 2004, Ektron went to a .NET-native solution with the release of version 1.0 of CMS400.NET.

In 2012, versions 8.02, 8.5 and 8.6 came out with focus on personalization and content targeting, Azure Cloud, and other improvements. Also in 2012, Ektron signed a mostly exclusive partnership with Baynote — a company whose quite esoteric “adaptive web

experience” and intent-based personalization technologies are used by companies like Dell and AT&T. (OpenText’s Vignette has an OEM deal with Baynote, as well.) While Ektron has yet to do something meaningful with this relationship, the intent is seemingly there for moving beyond simple business rules-based personalization techniques that are available within Ektron CMS.

In 2013, Ektron has set a rather active roadmap, releasing version 8.6.1 in January and 8.7 in April 2013.

Technology

Technical Administration and Security

Ektron CMS uses Microsoft SQL Server as a repository, but content — along with most system artifacts — is represented and managed as XML. It is, fundamentally, a coupled management and delivery system, which means that you use the same product to manage content as well as to serve it up to visitors. That’s not to say you can’t have physically separate staging and delivery environments, however. With Ektron’s optional eSync module, you can set up physically distinct environments (e.g., development, testing, and production) that can be separated by a firewall, with content deployed among them in any number of ways (e.g., single job, batch job, unidirectional, bidirectional, scheduled, or one time).

In short, it’s a “frying” system. In fact, there is no such thing as having the system completely bake an .aspx page into a static HTML page; Ektron CMS sees every page as an .aspx page at run time. Therefore, you’ll want to pay special attention to stress testing the product using your intended deployment topology, which in turn means you’ll need to spend some time fiddling with cache settings.

The eSync replication module allows Ektron servers to talk to one another and move content uni-directionally or bi-directionally. Replication of files, folders, system artifacts, and/or databases can occur on a scheduled basis, or in response to an appropriately privileged user’s real-time command. When synchronization occurs, only diffs (“things that have changed”) are sent over the wire, not entire database tables. Helpfully, the system lets you specify “exceptions” — items that should not be replicated as part of a sync — as well as things that should always be replicated (even if they normally wouldn’t be).

Because the number of configuration settings that can be applied to sync jobs is large, Ektron supports a notion of eSync Profiles (Figure 249), representing configurations that have been specified and saved in advance. Thus, when initiating a sync, an authorized user can choose among any available Profiles to apply.

Technology	
Technical Administration and Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

A consultant we talked to told us he was impressed with Ektron’s implementation of the eSync functionality. The traffic is encrypted and participating servers authenticate using certificates. Some caveats apply, though. First, source and target machines must both be running Ektron CMS; you can’t push content to a Windows server that’s not running the CMS software. Secondly, all traffic must go over port 8732 — you can’t use any other port. This is a relatively minor issue — unless your network administrators say you can’t use that port. However, not all users are happy with eSync. Some report this feature to be occasionally buggy — a common complaint for Ektron.

On the delivery side, all pages generated by Ektron are .NET pages.

Although user management is relatively straightforward, a couple of things are worth noting. For example, you need to delegate rights (manually) to users in order for them to perform certain mundane managerial tasks, like deleting or moving images and other files. You should plan on doing some investigating to see if your ideas about delegated administration match the implementation that Ektron offers. Some licensees have been known to dole out full privileges to department heads as a way to bypass some of the tedium involved in setting up individual rights. That’s obviously not a good practice.

Granular access control and delegated administration are further complicated by the way Ektron handles LDAP integration. Basically, you either integrate with Active Directory (to get automatic user updating, group assignment and administration, single sign-on, secure authentication, and other services), or you use a non-Microsoft directory and settle for LDAP as a means of authentication only. With LDAP, all users are in the “everyone” group and you lose some of the rich user-management features of Active Directory.

Ektron offers a reasonably broad set of managerial reports, including workflow reports.

Development

As with Sitecore, Ektron’s product is designed to support add-in “modules” on top of the core platform. Together with partners, Ektron is developing some industry-specific customizations for health care, higher education, and government. In fact, the product ships with the so-called “Ektron OnTrek” starter site with pre-built templates and pages for the most commonly used scenarios: home page, landing page, eCommerce, company pages, etc.

Vertical-specific “starter sites” also come as part of Ektron’s out-of-the-box install. These sites, which are functional sites minus true content, designed for particular use cases. For example, there are starter sites for Intranet, Legal, Travel, Publishing, Education, and Government. While the starter sites are helpful in getting off the ground quickly, buyers who

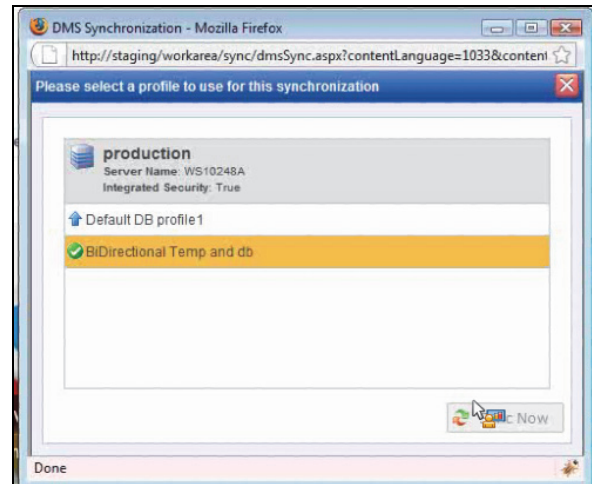


Figure 249. Synchronization configurations can be saved as “profiles” to be applied on-demand to individual sync jobs. The system offers many sync options and provides good reporting capability around replication.

plan on deviating from the layouts of these pre-packaged kits should be aware that modifying the layouts of these pages is not always as easy as making a simple CSS change. For best results, you'll need a .NET developer to modify the innards and change the corresponding templates.

Ektron also ships “Starter Applications” that run atop the site structure of your choice. These include a Photo Browsing app, a Wiki, and a Project Management module. The Project Management add-on provides SharePoint-like workspaces, albeit somewhat less intuitively. Of course, if you really want to use SharePoint, Ektron has a SharePoint connector, but the Project Management add-on is not designed with SharePoint in mind.

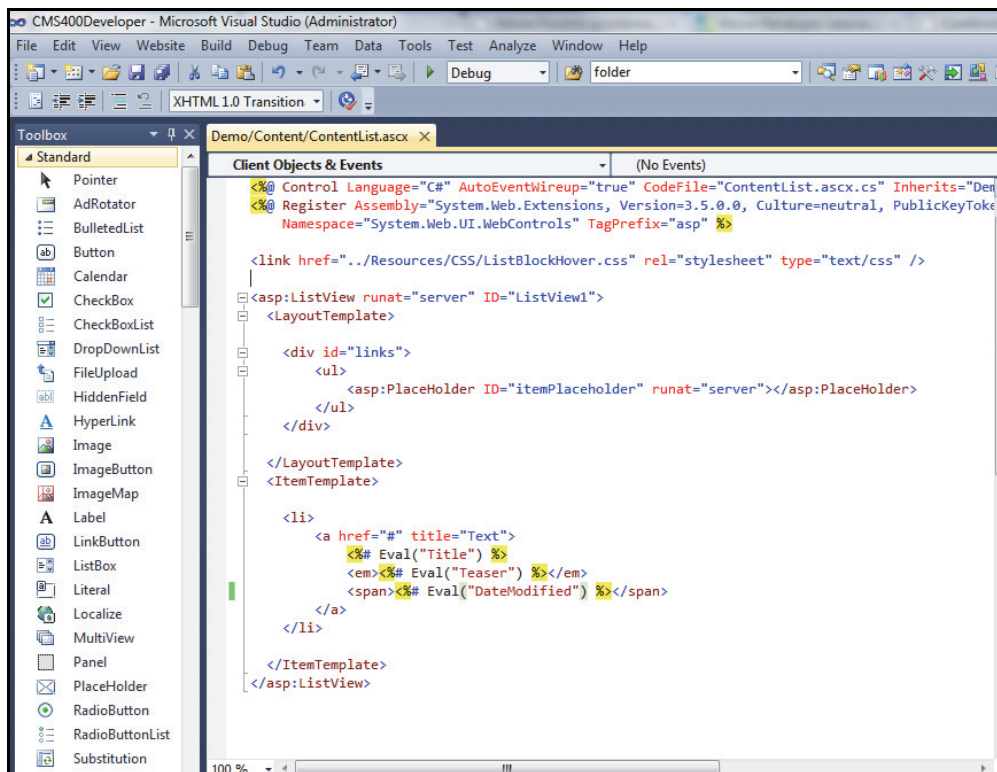


Figure 250. Ektron’s Visual Studio extensions allow you to drag and drop .NET controls.

The product’s standard tools are managed via browser, including administration of the system itself, although developers will want to avail themselves of Visual Studio, where the integration is quite deep. Ektron CMS ships with various pre-built components (e.g., for navigation, blogs, calendars, forums, and RSS, among others), which are all accessible from Visual Studio or Dreamweaver plugins, where they can be dragged and dropped from an explorer-tree pane right onto pages (Figure 250). This certainly simplifies template development, but it’s not something you’ll be comfortable with if you’re not a developer.

Ektron CMS natively employs XML for content management. To get the most out of the system, your developers or consultants should be on a first-name basis with XML, XPath, XML Schema, and XSLT. Keep in mind that developers accustomed to dealing with SQL may find traversing XML to be more difficult; they have to write their own XPath queries. Your *authors* do not have to be subject matter experts in these areas (they can do everything they need to do in the UI shown in Figure 251), but even they cannot escape the constraints imposed by structured authoring.

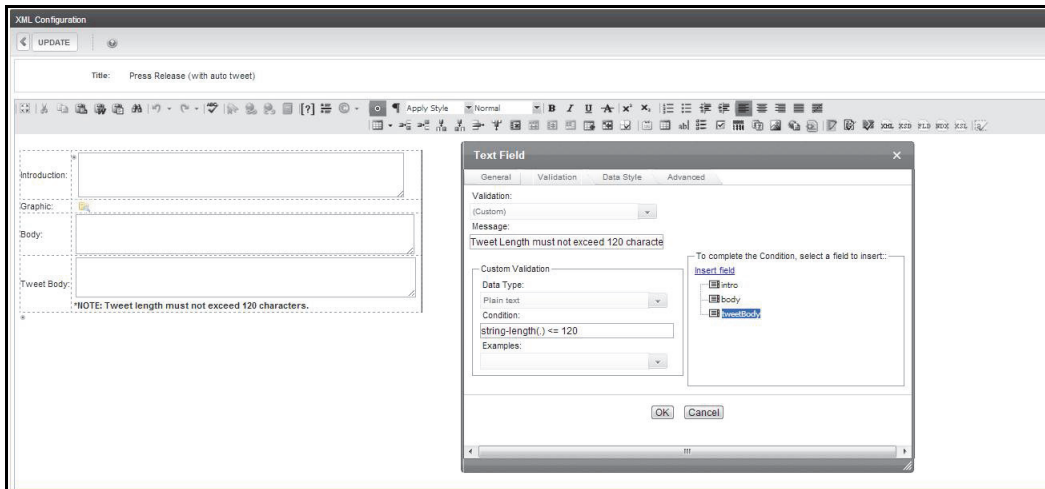


Figure 251. Ektron can present the content contributor with a structured-content interface (above), an in-context interface, or a rich-text-editor interface, depending on the user’s role and the system configuration.

Ektron calls content types “XML configurations.” A developer creates a base configuration for a content type and binds it to an XSL template created offline. The developer then applies multiple XSLTs for content output — one for each rendition. You can apply that same configuration for different areas of the site, but if the output needs to change in different directories, the author needs to know which XSLT source to select. Moreover, you can control which CSS file is applied at the item or folder level, so you can have the same object appear in different ways in different parts of the site.

Menu selections, navigational link collections, and other UI items can be customized through the so-called Smart Forms functionality. In many cases, a power user (rather than a programmer) can configure a “Smart Form.” The form becomes a structured-authoring template for content contributors; meanwhile, the entered content can be pulled together dynamically at page-request time with XML fragments and embedded (element-level) queries.

Smart Forms can be integrated with Flex technology to create highly dynamic, visually rich web pages. We saw a compelling demo, but you’ll want to test it in real-world scenarios. In any case, it’s indicative of Ektron’s broad technological ambitions.

Developers can also take advantage of Ektron’s integrations with Microsoft SharePoint to create portal-based content management screens. In 2013, Ektron expanded its SharePoint integration capabilities to the point where it is possible to manage some metadata and import it from SharePoint into Ektron through the mapping UI. Because this essentially creates two sets of metadata, it is important to understand how to maintain them. Any changes to metadata done in and published from SharePoint will automatically reflect in Ektron via Windows Service, but there’s no notion of bi-directional exchange of data. Thus, none of the data changes that occur in Ektron will be reflected in SharePoint.

With the introduction of the new Framework API in 2010, developers working with Ektron can learn a few new tricks behind the Ektron.Cms.Framework namespace. Consistency and discoverability for performing CRUD operations were the main drivers behind introducing the new API. It also includes a partial implementation of Microsoft LINQ (Language-Integrated Query) for filtering and sorting.

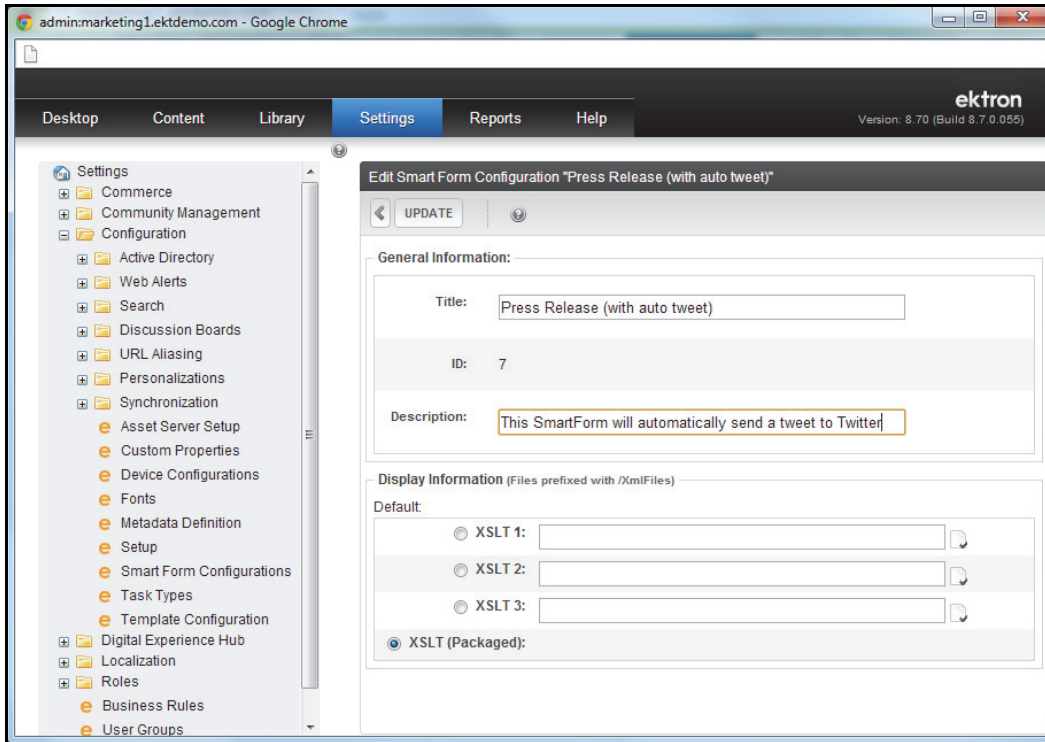


Figure 252. The SmartForms form-creation interface is no longer tied to ActiveX.

In theory, you may no longer need XPath or XSLT skills developing on Ektron under the new API, since data can be accessed as objects, in line with more pure .NET development. In practice, the API is new and probably under-tested in the field. Wait for others to skin their knees.

Among other things, while Ektron’s goal is to have every object in the CMS available through the API, they’re not there yet (e.g., menus and collections are not yet API enabled). Also, you can only sort, list, and group in this iteration of the Framework API. You may want to investigate further within the Ektron developer community, which is becoming more active on the forum and at live, local user-group gatherings.

On the standards side, Ektron quietly gave up on its CMIS implementation, which used to be “in beta.” According to the company, it went 80 percent of the way before halting, due to the reported lack of customer interest in this interoperability specification. Ektron’s repository thus remains manly proprietary, surrounded by .NET-friendly — but proprietary — APIs.

As Ektron continues to further develop and change its APIs, some developers run into issues every now and then due to the changes. For example, you may want to consult the latest documentation to make sure that namespaces that were included in prior versions are still included as part of the updated framework.

Performance

Traditionally, most customers have run the system and their website(s) off a single box, but — as is true with most coupled architectures — performance under load can be a concern, particularly if intricate personalization is necessary. Remember, this is a frying system that

relies on dynamic page assembly at request time. You'll want to test carefully if you anticipate having to serve high volumes of highly personalized content — especially, in light of Ektron's renewed love for expanding personalization and content targeting capabilities.

Content

Contributor Experience

Before Ektron became a WCM vendor, the company was focusing its efforts on developing a rich text editor. Not surprisingly, up until recently you had a choice between the venerable eWebEditPro+XML, or the newer, AJAX-based eWebEdit400. The latter will run in Mac or Windows environments, on Firefox or IE, and has no ActiveX dependencies. The former still uses ActiveX, but it is not a required component anymore; according to Ektron, the new AJAX UIs have 100-percent feature parity with the older ActiveX technology, and the older option is still there for backward-compatibility purposes only.

Starting with version 8.7, Ektron switched to the Aloha rich text editor as its default. This editor came from an OEM agreement with Austrian CMS vendor Gentics, which spearheads Aloha development and support. In prior releases (8.6 and 8.6.1), there was an option for Aloha as a separate install. Some of the reasons behind switching to Aloha are that it's lighter and often faster, and it also offers HTML5 support.

Content	
Contributor Experience	
Overall Usability	●
UI Accessibility	●
Contributing Content	
Authoring & Transformation	●
Tagging & Taxonomy	●
Content Reuse	●
Media & Document Management	●
Repository Services	●
Content Lifecycle	
Workflow	●
Globalization	●
Archiving & Compliance	●

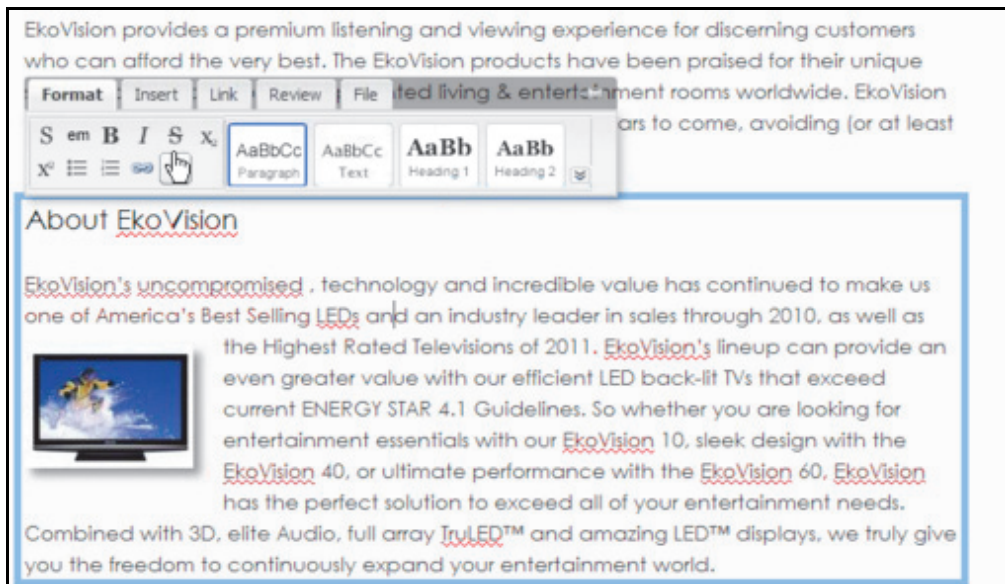


Figure 253. Aloha is the newest default rich text editor in Ektron CMS.

On the configuration side, it's fairly easy to make a switch to Aloha by changing third-party editor settings. If you're a current Ektron customer, it may make sense to test Aloha in your own testing environment before making the switch.

Contributing Content

Ektron’s Page Builder UI (Figure 254) gives non-technical users an easy way to build gadget-rich pages through drag-and-drop gestures. At the top of the window is a fly-out “gadget bar” that stays hidden most of the time; you toggle its appearance, as needed. You can drag gadgets (representing things like RSS feeds, Facebook widgets, Google gadgets, etc.) onto the page and very quickly build a dynamic web experience. Pages can be workflowed, previewed before going live, and cloned and modified to support A/B testing.

While it will not directly import and transform desktop documents, the product can recognize and convert the formatting from Word documents upon copy and paste, and does a decent job of it. Also, you can open and edit Word files directly within the browser.

More generally, content reuse — as opposed to basic document repurposing — remains under-exploited, although the company has made some improvements at the behest of some of its larger customers. Ektron has a notion of a “searchable category,” for site-wide information discovery. You assign category types to folders, and after content is tagged, developers can write canned search routines that can drive metadata-based (faceted) navigation.

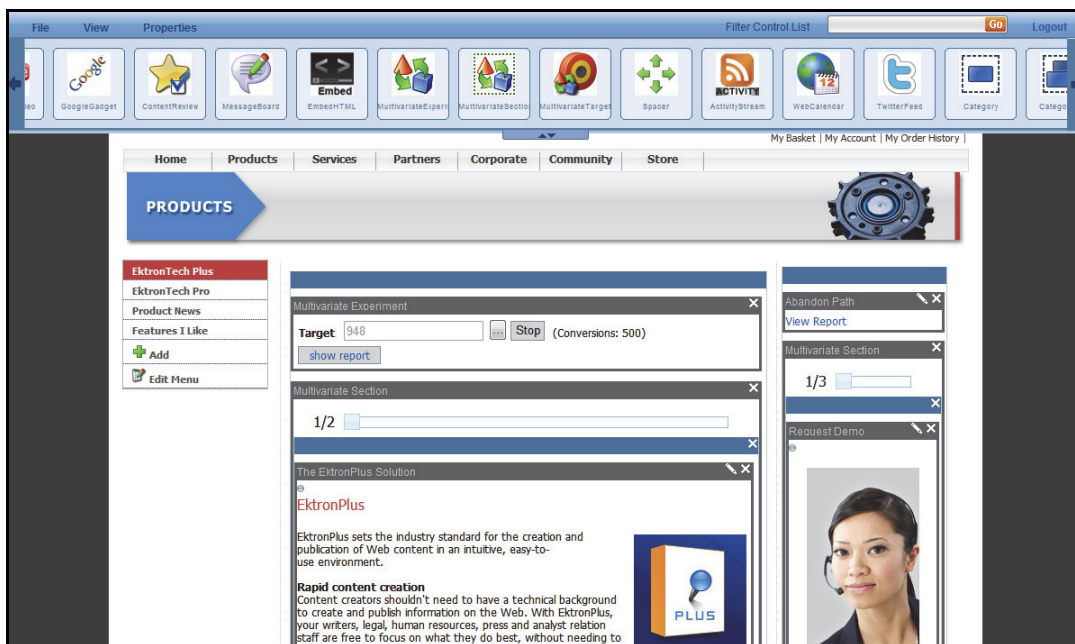


Figure 254. The Page Builder UI places a fly-out menu bar and gadget bar at the top of the screen (easily hidden with a single mouse click). You can drag-and-drop widgets onto designated drop zones on the page to build a page quickly.

Ektron users tend to like the product’s form builder. Non-developers are able to build forms easily and quickly. Developers like the fact that the forms themselves are stored as content, as are the three options for what to do with submitted data; send it to an email address, login to a database, or send it to a custom form handler. These are good submission options, as long as you only want to choose *one* of them. If you want your forms to log to a database *and* send an email, more development effort is required. However, some users would like to see some common business logic options get built into the system.

It was not until version 7 that Ektron introduced the much sought-after functionality of taxonomy management, which allows users to create and manage taxonomy categories with a minimum of fuss (Figure 275). You can right-click on a term and edit, add, reorder, or delete it.

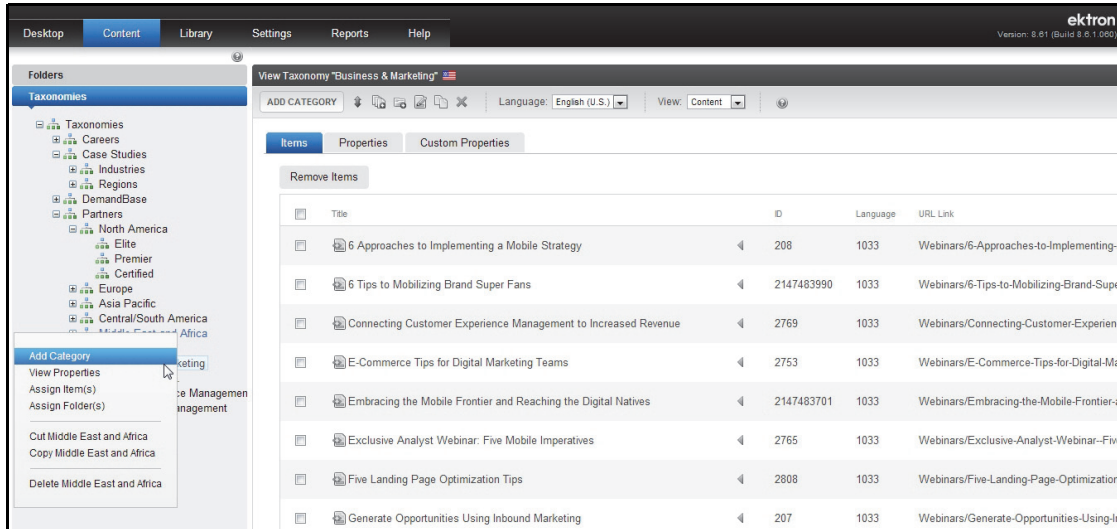


Figure 255. Ektron's new taxonomy management interface.

Also on the metadata front, Ektron has a feature called geomapping, otherwise known as integration with Google Maps. For designers of commercial sites that steer customers to retail (or other) locations as part of a search session, this is deemed essential. Community-driven sites and others have found numerous uses for geomapping and map-based query.

The company tends not to embed third-party code, and true to form, wrote its own Microsoft-flavored search engine when the need for such a facility became apparent. However, search can be notoriously tricky — and resource intensive — so you'll want to monitor performance closely, particularly if you are indexing on a large number of fields. Test carefully.

Ektron's homegrown site search capabilities are based on integration with Microsoft Search Server 2010, with Microsoft FAST Search as the next available (more sophisticated and less homegrown) option. Elements can be set as unsearchable, either by default or on an ad hoc basis by the author. Using an "XML Indexing" feature, you can create parametric search capabilities, and there is underlying support for regular expressions for power users who are brave enough to care. One of the main reasons you may lean toward Microsoft Search is its ability to load balance and is not limited by a cap of three-million indexing items.

For asset management, the product ships with a basic image library capability, though you may find it a bit unwieldy compared to other vendors' offerings (Figure 256).

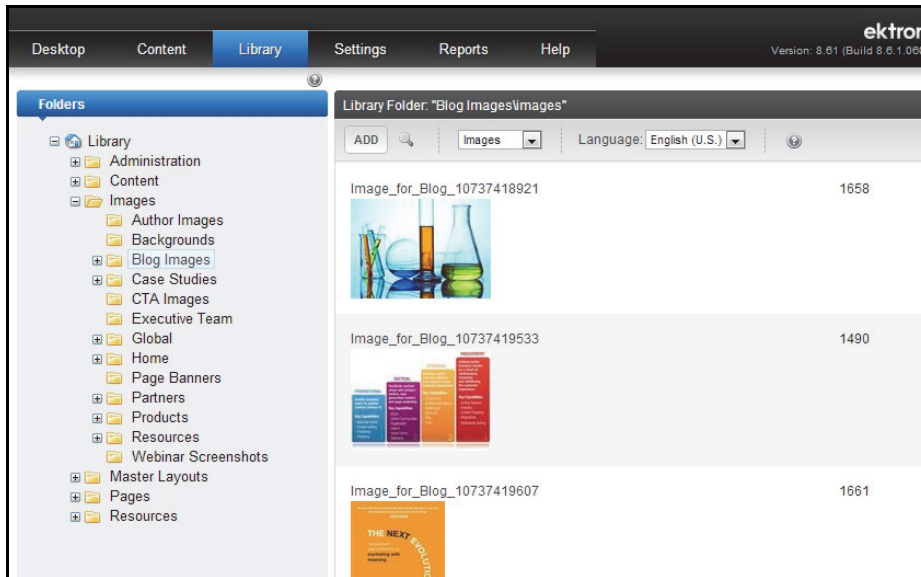


Figure 256. Ektron’s image library is functional, if not DAM-like.

Content Lifecycle

Workflow is, typically, stronger from an engineering standpoint than usability perspective. The product ships with a task-based workflow engine with strong reporting capability. It allows for parallel workflows and the creation of tasks without a content payload (many costlier tools cannot do this), and the latest version allows you to categorize and prioritize your tasks. Editors can collaborate via attached forums and see visual differences of recommended changes. Overall, the workflow infrastructure is quite strong.

User interfaces are another matter. One problem is that the task inboxes are not particularly helpful or descriptive (Figure 257), and the workflow interfaces seem a bit difficult to navigate. This is another example of a UI that may not scale well for someone who is actively involved in many different types of tasks. Some customers have complained about inflexibility in managing and clearing tasks. In theory, all of these things can be fixed with enough custom development work, but the point is, if your processes are heavily reliant on complex approval workflows, you’ll need to investigate the suitability of Ektron’s workflow system quite carefully as you evaluate your organization’s needs. Investigate it not just from the perspective of requirements, but also from an actual usability point of view.

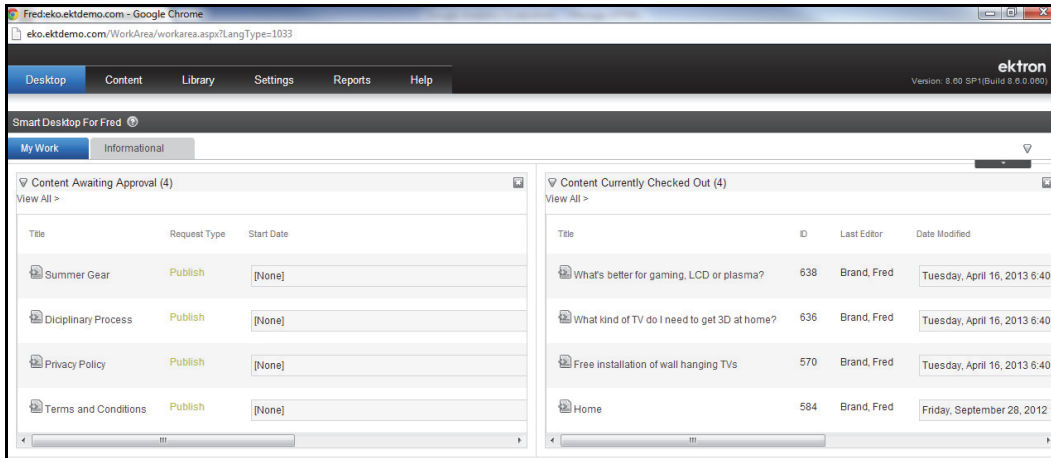


Figure 257. Ektron's contributor dashboard can be configured to show workflow approval status (left), and Content Currently Checked Out.

Ektron has worked hard to improve the product's globalization capabilities, but it is still somewhat weaker than its competitors like Sitecore or EPiServer. The XML capabilities of Ektron come in handy here. A single content instance can reference multiple localized children, but they all remain within the same instance and are not easily managed, browsed, or recognized. Content items, for example, can be enabled to have children in other languages, but the awkwardness of multiple drop-down menus with language selections is there and shouldn't be overlooked if multilingual content management is vital to you. In both the contributor and visitor interfaces, you can toggle back and forth between approved languages within the same interface (Figure 258). Some users find this handy, and others find it confusing, due to the lack of clear separation among the different language versions.

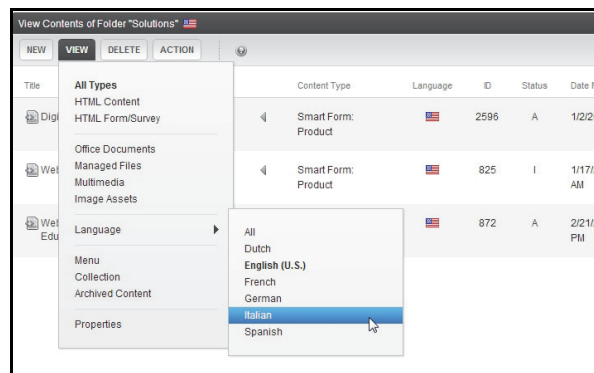


Figure 258. Toggling between languages.

Similarly, the product supports localization workflows, either through the new workflow engine, or via export and import of industry-standard XLIFF XML files. The former can still feel a bit manual and authors may need to keep temporary track of content ID numbers. The latter approach is also somewhat clunky, since there are few browser-based screens to exert finer control on the export and import of XLIFF files. Nevertheless, a product that was heretofore suited only to the North American marketplace can now be used to support multinational publishing efforts.

Experience

Publishing

With respect to friendly URLs, the product now has a convenient automatic aliasing feature that can be configured per folder or per “content block.” You can specify a URL-construction pattern that’s either folder-path based or taxonomy based, or (if you’re geeky enough) regular-expression-based. Using the taxonomy method, you have to be sure to instruct users not to use the same name for two different content items that might resolve to the same taxonomic path, even if they’re in different directories: Beware. The end result is human-readable URLs in lieu of.aspx URLs with funny parameter chains.

For mobile support, Ektron offers mobile-specific templates and device detection using the WURFL (Wireless Universal Resource File) framework. While this mobile offering is not nearly as elegant or bug-free as Ektron’s rivals, it’s a good start if your WCM plans are not overly ambitious and include the ability to publish content to the mobile channel in a properly managed and rendered way. Ektron allows you to build and apply mobile-specific templates and apply them either to content that’s been written for the web, or to the content created specifically for mobile devices (usually managed from a separate “mobile” content folder in the work area). As with other vendors, these services rely on an open source WURFL framework, which essentially outputs an XML configuration file with available and supported mobile device models.

The Targeted Content widget helps you target-specific content to specific devices if the conditions are met. One of the conditions is device detection. The system can detect the site visitor’s device and render content applicable to the specific device. The device-to-content mappings and configurations are done manually in the administrative area of the UI.

What is clearly lacking is a clean emulation environment, which enables preview capabilities for marketers and content editors to get a good view of how the content will render on a specific device, which is very similar to its issues with web pages, in-context editing, and previews on a staging server. The shortcoming may delay mobile technology adoption among Ektron users. If mobile is an important channel for you, do your due diligence in homework and testing before buying Ektron CMS. While this mobile offering lacks some of the elegance and stability you might find in competing offerings, with enough time and effort, you could probably extend it to meet your needs.

In recent releases, Ektron extended its Ektron Cloud Manager support from Microsoft Azure Cloud to include Amazon EC2. Although there are not many customers actively using this offering, it could prove to be worthwhile for organizations looking to supplement (or even replace) their current infrastructure with the wizard-based auto provisioning of Ektron sites in the cloud-based content delivery environment.

Experience	
Publishing	
Standards Adherence	
Multichannel	
Mobile	
Digital Marketing	
Site & Campaign Analytics	
Testing & Optimization	
Segmentation & Personalization	
Social Media Integration	
Promotional Campaigns	
Community & UGC	
Workplace	
Collaboration & Networking	
Dashboard	
Ancillary	
Site Search	
Online Forms	
Module Ecosystem	

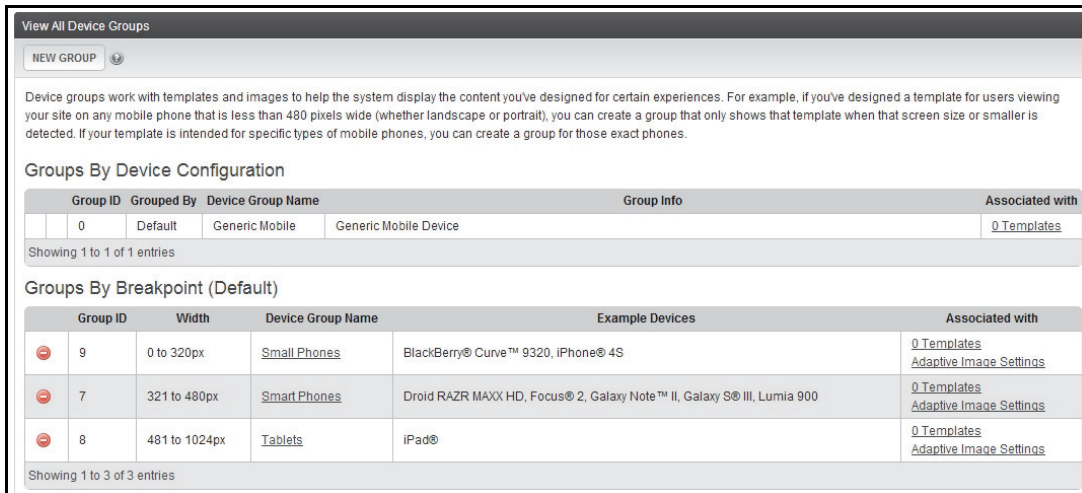


Figure 259. Device configuration view in the UI: Simple actions like adding or removing a device model from a pre-configured library of devices, and setting width and height parameters. The basics are there, but not for sophisticated use cases. “Feature-hungry,” and “execution-incomplete” are not rare descriptors of Ektron’s product development practices.

Digital Marketing

Ektron offers a fair bit of WCXM functionalities here, though the offering may feel more wide than deep. First, the company’s SEO Site Checker utility provides reports for each web page, including information on metadata, compliance with best practices around structuring a page for SEO, and a keyword density analysis.

Ektron’s analytics and reporting capabilities have improved — mostly by dint of jettisoning its own home-grown analytics in favor of integration capabilities with Google Analytics, Webtrends, and Omniture. Reporting is integrated in the UI at the standard report level. You should cautiously navigate this area of the product, though. Despite the general approach to deprecating proprietary analytics, Ektron’s WCXM system still has some of those leftovers under the umbrella of “Business Analytics.”(Figure 260).

In 2012, we saw the debut of Digital Experience Hub (DXH), which is a separate offering that specifically targets organizations looking to manage online customer experiences in tandem with managing websites using Ektron CMS. In its current state, DXH is mainly a set of pre-built connectors to popular digital marketing third-party applications such as HubSpot, Salesforce, and Marketo. The analytics capabilities we review above can also fall into DXH, or can be accessed from WCMS if you do not have a DXH license. Keep in mind that the definition of “connector” in this case can vary very widely; some of these integrations are tighter than others, and the depth of what you can accomplish with the HubSpot connector will vary from the Marketo connector, for example. In any case, be prepared from a resource and budget standpoints to invest more into DXH and to customize these connectors to leverage their digital marketing capabilities in tandem with Ektron.

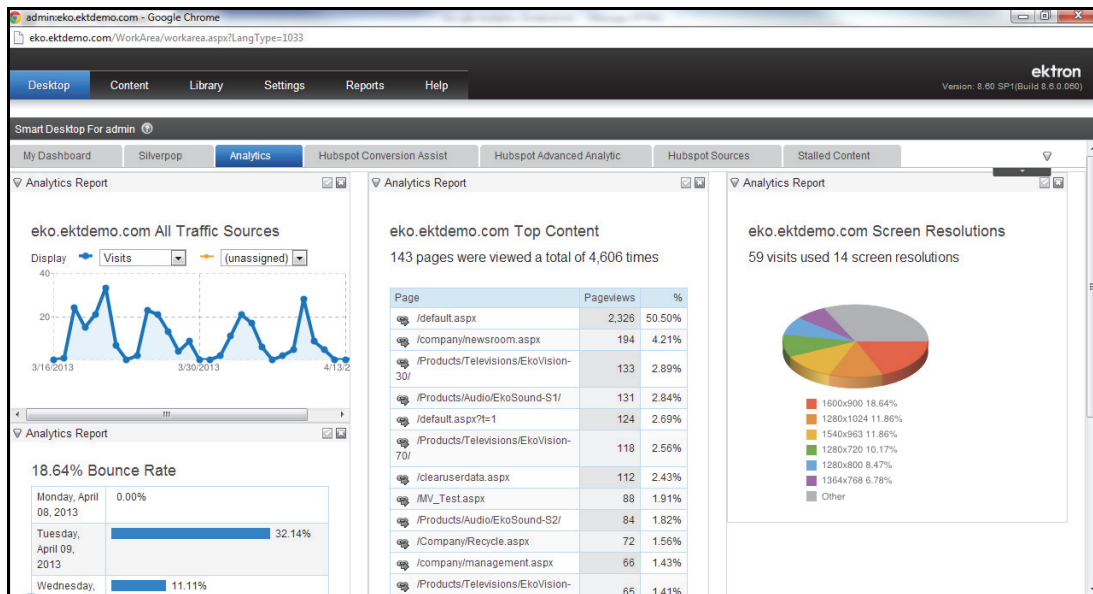


Figure 260. Ektron provides an integration with Google Analytics (unlike Sitecore, which has created its own analytics services (a mistake, we think).

Ektron CMS has a slew of micro-applications, including polls, surveys, blogs, and threaded discussions with permissioned commenting, among others. The product also comes with its own email subscription/marketing system — something customers typically procure via a third-party provider.

Larger customers may want to procure these services elsewhere, since Ektron’s capabilities here aren’t as good or as flexible as what you can find elsewhere on the market. On the plus side, they are immediately available to the customer who has *not* gone elsewhere for this extended functionality.

In the latest version of Ektron CMS, the company has added enhanced personalization features. One of the more interesting developments is “Social Targeting,” included in Ektron's Marketing Optimization Suite. This functionality is designed to create and deliver personalized and targeted content, based on site visitors’ Facebook profile data, social graph, and related attributes exposed via Facebook’s Graph API.

Non-social personalization revolves around a cocktail of scoring, based on a point system, navigation path, business rules (defined by marketers), GeoIP lookup, and behavior. While there’s no official integration, there are talks about being able to export segments from Ektron as leads into Salesforce marketing tool, as well as set values in Salesforce for content targeting on the web.

In a possible attempt to move beyond this simple, business rules-based personalization technique, Ektron has reached outside of the company, signing a partnership with “adaptive web experience” company Baynote. It is too premature at this stage to evaluate any strengths or weaknesses of this move, and even the vendor could not sufficiently put a finger on how the integration would work. No current Ektron customers are using this functionality.

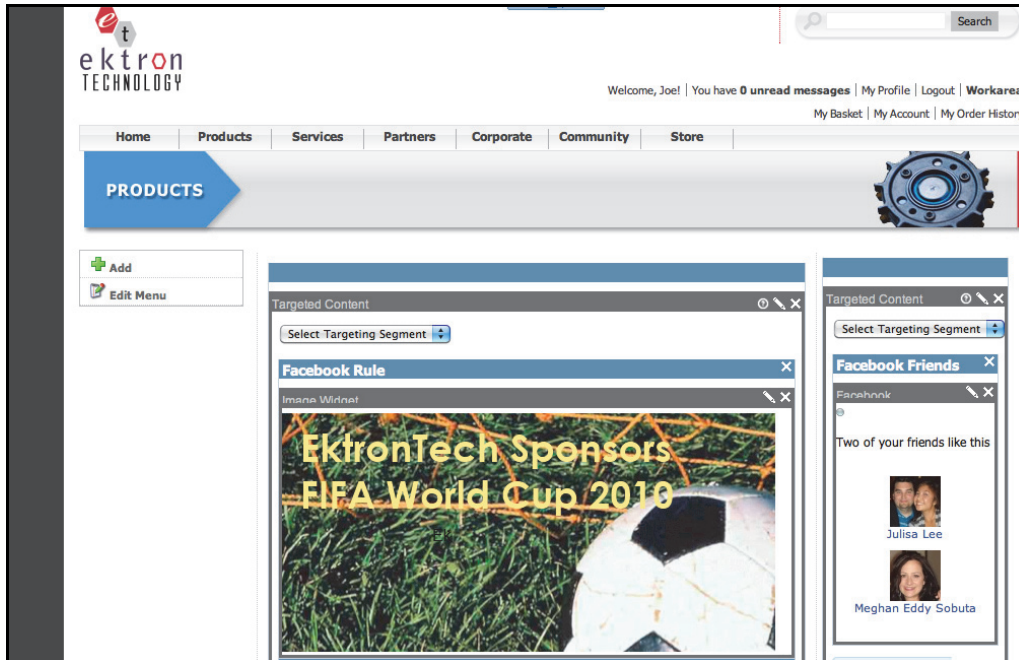


Figure 261. Managing segments and associating content targeted to those segments in Ektron's PageBuilder.

Ancillary Services

True to form, Ektron provides its own search service or (preferably), you can apply a third-party engine.

Ektron's bevy of extended services becomes a strength of the platform — if you prefer that all of these services come from your Web CMS provider. Keep a close eye on Ektron's recent trend to get involved in more and more partnerships in search for OEM'ing the missing functionalities — albeit Baynote for personalization, Brightcove for video asset management, or Lionbridge for translations. While the intention appears to be good, the many integration points could be difficult to manage, depending on how well or how poorly they're executed.

Vendor Intangibles

The Ektron system is complicated enough that you may want to avail yourself to outside expertise when treading beyond the limits of the pre-canned Starter Sites and Starter Apps (although many customers never do). Once upon a time, Ektron relied on a large stable of resellers and consulting partners around the world, but in recent years, it has beefed up its own sales,

services, and support-engineering resources — to the point where more than one Ektron partner we've spoken with recently has voiced concerns about channel cannibalization. Ektron says it's had to bail out its channel partners too many times — a common vendor grievance, but still not a good sign. Now with 80+ employees in Professional Services — and its biggest

Intangibles	
Vendor Services	
Vendor Professional Services	●
Channel Partner Services	●
Support & Community	●
Strategy & Roadmap	●
Viability & Stability	●

customers served directly by the vendor — it's safe to say that Ektron is its own biggest implementer.

This has two ramifications. First, if you go the partner route, carefully assess that firm's skills and long-term relationship with Ektron. Second, if you are looking for ancillary design or integration services that you would typically obtain from an agency or SI, you will likely have to work with another company as well as Ektron. Contrast that approach with Sitecore's, EPiServer's, and others, where the agency is also your CMS implementer, and the vendor stays at arm's length. Neither approach is perfect; just know which one will work better for you.

In any case, the company's first-line of support is well regarded by many and known for having a consulting mind-set, perhaps from years of coaching lone developers at small companies about how to install and optimize the product.

Even today, Ektron (perhaps due to its affordable pricing for a single site) tends to appeal to the lone do-it-yourself developer and/or the one- or two-person Microsoft IT shop. Ektron's DevCenter <http://developer.ektron.com>, speaks to this customer base, and apparently does it quite well. Ektron has invested in expanding and enriching the DevCenter site over the past several years, with good results. The company also claims to have trained more than a thousand developers via its developer certification program.

The company site exposes direct links to product documentation, including developer docs, on the public site — all freely downloadable with no need to pre-register for access. This is in stark contrast to the vast majority of CMS vendors that zealously guard their documentation.

As previously mentioned, Ektron rolls out new features and point releases at a dizzying clip — which is in some ways commendable, but has also caused customers to wonder at times if they were doing Ektron's QA work for them. "Feature driven development" of the type Ektron has engaged in is indicative of an engineering-oriented culture that tends to value technology and gee-whiz features over more mundane things like human factors testing. On the other hand, customers also praise the company for its receptive attitude toward feature requests and feedback, in general.

We've seen Ektron grow from a couple dozen employees to over 200 (despite periods of downsizing). Now the company has offices in Australia and the UK, as well as — more recently — Toronto, Canada, and Austin, Texas, and San Francisco, California in the United States. Nonetheless, the company somehow retains a "small-business," even cowboyish-like feel — which you may or may not appreciate.

Some customers praise Ektron's lack of formalities and relatively easy access to senior staff. Others worry about the lack of formal structures needed to oversee what has become a much different company than it was as recently as five years ago — on a global scale. However, at present, customers report generally good access to support staff and fast response times.

Ektron's pricing — while having changed slightly — remains reasonable for what you get, but seems to punish customers with multiple websites. Ektron CMS starts at US\$15,000 for 1 URL, 1 server and five users; 10 users - \$20K, and 25 users - \$40K. However if you want to go enterprise level (as many Ektron customers do), your basic entry fee is \$100K; this is server-based pricing for unlimited URLs for one authoring instance and two delivery servers. Costs go up as additional displays or authoring servers are added.

eMobile Tools, Social Software, eSync, eCommerce, Marketing Optimization Suite (with functionalities like MVT, analytics, content targeting, SEO). The Digital Experience Hub is sold on top of the base license at 20 percent of that license amount each. So, at the enterprise level, you should expect to pay US\$20K for each additional module. The typical standard (i.e., 8-5, Monday through Friday) annual support and maintenance fee runs at a common in the industry 20 percent of the total license fee. When budgeting your Ektron CMS purchase, don't forget to add training costs on top of the license, as well as the actual implementation costs — either via Ektron's Professional Services or a partner.

While the pricing schedule is somewhat complicated and (at this point) arcane, Ektron does try to accommodate every class of customer.

Conclusion

Ektron could present a decent technical fit for almost any size Microsoft shop, even those with severely limited IT resources. Indeed, this system tends to cater to the do-it-yourself developer that can extend and customize the CMS, rather than advanced .NET development teams. Not surprisingly, some customers look at Ektron from a “package” perspective and find its use mainly in uncomplicated publishing and basic interactivity.

Still, larger enterprises should note that Ektron has put a great deal of effort into making Ektron CMS more scalable, offering a powerful synchronization module (along with a load balancer) and ample best-practices advice around clustering and scale-out.

If you stand back and try to take it all in, you'll get a picture of a CMS vendor that tries hard to offer a self-contained “everything platform” — the Swiss army knife of WCM systems — with a multifaceted pricing and licensing scheme to match. Need comprehensive site search? Ektron bundles it. Geomapping? It's there. Wikis? Built in. Ditto for a variety of other social networking features. Indeed, it's unusual to have so many apps and features rolled up into one product offering. You just need to recognize that Ektron tends at times to favor features over usability and stability. As one of Ektron's customers says, the company is “feature-hyper,” with new functionality getting built into the product very quickly. This too often results in poor control over release processes: going to GA too early and letting the bugs out, followed by hefty service pack releases; as well as releasing a myriad of server controls (there are about 90 of them shipped with the product) that are too surface-level and not very useful without customization.

Depending on your needs, you may find far fewer integration hooks into other applications than you'd find in a product like SDL or Sitecore. If Ektron's ample out-of-the-box features meet your needs, you'll find it a comparatively advanced — albeit wide, rather than deep — product for the price, similar in some ways to EPiServer. At times, Ektron is chosen over SharePoint for its considerable pricing advantage, especially for intranet scenarios, but be very clear about your use cases and scenarios if you're measuring the two against each other. If you need to perform significant customization or integration work, you may find that competitors can offer cleaner development alternatives.

In the meantime, if price is a primary consideration, you may want to consider Kentico or Telerik. Neither product can match Ektron's feature set and maturity in the market, but both are likely to cost considerably less.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Enonic: CMS

enonic.com

Project at a Glance

Specsheet	Enonic: CMS 4.6 Summary
Geography	Primarily Norway and the rest of Scandinavia
What's New	<ul style="list-style-type: none"> • Improved versioning and page "snapshot" capabilities • Disk-based binary store replaced database storage • System performance tracing tool • Friendly URLs generate as standard • "Assign" workflow feature replaces previous one-step system
Strengths	<ul style="list-style-type: none"> • Leverages standard J2EE technologies familiar to most Java developers • Broader-than-average support for databases and appservers • SOAP-RPC API, in addition to pure-Java API • Relatively low learning curve for authors and approvers • Good value: decent feature set for the price • Company and product very straightforward: the antithesis of "slick"
Weaknesses	<ul style="list-style-type: none"> • Highly XSLT-centric, which means a steep learning curve • cursory documentation awaits developers who wade into system • Permissions management is potentially tedious • UI is oriented toward techie contributors rather than marketers • Coupling between menu and documents is complex • Workflow capabilities are primarily ad hoc in nature • Surprising native absence of support for globalization scenarios • No integration with Microsoft Office • No out-of-the-box blogging, wiki, or collaboration apps • Not well-known outside of Norway
Potential Fit	Microsites & Landing, Multichannel Publishing, Basic Digital Marketing
Unlikely Fit	Advanced Marketing Portal, Digital Workplace, Global Enterprise, Ultra-Large Single Site
Compare To	CoreMedia, EPiServer, Escenic, Oracle, Magnolia, e-Spirit
Operating Systems	IBM AIX, Windows, Linux
Repository	Database: Most RDBMSs supported, including MySQL 5.1 and later
Client	Browser
App Platform	Java application server: Tomcat, BEA WebLogic, IBM WebSphere, JBoss, Oracle Application Server, Sun AS, Glassfish
Licensing	Enterprise starting at €16,000. Community Edition under open source AGPL 3.0 license
Ownership	Private — majority owned by management and private investors

Summary

Enonic CMS is used mainly in Norwegian organizations for intranets, websites, and extranets. The company has a loyal, long-standing local customer base, which has kept Enonic well-funded during its continuing effort to expand beyond Norway.

Like neighboring Escenic, Enonic CMS is a Java-based system, and as such, it tends to follow standard Java EE architectural paradigms, which, for Enonic, now also means running inside the Spring Framework. What this means for customers is that if you already have significant Java expertise in-house, and if you already employ a Java-based portal, you may very well find that Enonic CMS is a good match for your IT organization.

On the flip side, if your technical personnel come mainly from the LAMP camp or don't speak anything but .NET, the developmental learning curve will be potentially quite painful. Either way, there will be pain involved, if you don't currently have any XSLT experts.

What about your business users? They may welcome a very straightforward, non-slick system that does the basics well, for a fairly low price. In our experience, however, the system works best for smaller teams of more technically oriented or "power" contributors.

Scenario Fits	
Simpler Site	
Informational	<input type="radio"/>
Microsites & Landing	<input checked="" type="radio"/>
Mid-Range	
Basic Digital Marketing	<input checked="" type="radio"/>
Mobile Site	<input checked="" type="radio"/>
Community Oriented	<input checked="" type="radio"/>
Complex Site	
Advanced Marketing Portal	<input type="radio"/>
Global Enterprise	<input type="radio"/>
Multichannel Publishing	<input checked="" type="radio"/>
Ultra-Large Single	<input checked="" type="radio"/>
Digital Workplace	
SMB/Departmental	<input checked="" type="radio"/>
Global/Enterprise	<input type="radio"/>

Introduction

Founded in 1998, Enonic succeeded in winning some marquee names, but mostly in its Norwegian home market. In 2006, Enonic opened an office in the US in an attempt to broaden its geographical horizons, but the company landed comparatively few customers in North America, and says it has toned down its operations there for now.

Similar to Danish vendor, Sitecore and Sweden's EPiServer, Enonic implements projects through partners only. This may make it easier for Enonic to win over new channel partners, but it may also make projects more expensive as a result, due to increased implementation costs and the presence of "another throat to choke" in the mix.

Feature wise, Enonic CMS releases have been evolutionary rather than revolutionary. Version 4.4 brought enhancements for mobile support; version 4.5 introduced improvements to workflow and audit control (versioning, rollback/forward and time-based "snapshots").

Perhaps the most notable change in Enonic's approach of late has been in licensing. Starting with v4.6, there are now two ways to license the product:

1. Community Edition – Distributed as an open source project via GitHub under AGPL 3.0, with a slimmed-down feature set
2. Enterprise Edition – This version has additional functionality and product support from the vendor

This licensing model renders the Community Edition product often lacking an enterprise flavor, which is something to watch when evaluating Enonic — specifically from the roadmap perspective.

Technology

Technical Administration and Security

The product has a Java architecture and runs as a web application with an application server. Enonic CMS's server-side architecture is based on the open source Spring framework, a well-established runtime-services framework that allows (among other things) component interactions to be managed via small XML configuration files.

Enonic CMS publishes dynamically in “frying” mode. The product comes with a handy Publishing Wizard, which takes the editor through a set of steps to decide how the content object(s) should be published. While the wizard is visually nice, a few licensees report that keeping environments (staging and production) in sync requires careful planning, as the wizard is focused on content, but not necessarily templates and other artifacts.

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

Previous Next Cancel

Step 2 of 3: Publishing

Description
 Choose page template and sections. Choose autoapprove for quick publishing. Home determines where the content belongs on the site.

Enonic

Page template: Preview

Select	Manually order	Home	Section
<input checked="" type="checkbox"/>		<input type="radio"/>	/ en / home / company / events and training
<input type="checkbox"/>		<input type="radio"/>	/ en / home / company / events and training / past events and training
<input type="checkbox"/>		<input type="radio"/>	/ en / training / content manager / Courses
<input type="checkbox"/>		<input type="radio"/>	/ en / training / developer 1 / Courses
<input type="checkbox"/>		<input type="radio"/>	/ en / training / developer 2 / Courses
<input type="checkbox"/>		<input type="radio"/>	/ en / training / developer 3 / Courses
<input type="checkbox"/>		<input type="radio"/>	/ en / training / hosting / Courses
<input checked="" type="checkbox"/>		<input checked="" type="radio"/>	/ en / training / site manager / Courses

Publisering.no

Page template: Preview

Select	Manually order	Home	Section
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="radio"/>	/ forsiden
<input checked="" type="checkbox"/>		<input checked="" type="radio"/>	/ opplæring / Kurs
<input type="checkbox"/>		<input type="radio"/>	/ tor

Previous Next Cancel

Figure 262. The Publishing Wizard.

Some customers have opted to replace the delivery engine with their own rendering or third-party systems, facilitated by the Web Services layer in Enonic CMS. However, Enonic suggests that the vast majority of contemporary customers are using their own built-in portals.

Morten Ø. Eriksen | Change password | Support | Documentation | Log out

Sites / Enonic / Menu items / en / home

Type:

General Properties Page configuration Security

Security

	Read	Add	Create	Publish	Update	Delete	Administrate	
Administrators	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="text" value="✕"/>
anonymous (anonymous)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text" value="✕"/>
Enonic Employees	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="text" value="✕"/>

Add

Save Cancel Preview

Figure 263. Access rights can be managed on a group basis via a simple, but functional interface.

There is no notion of “roles” *per se* within Enonic CMS. The product comes with five predefined group types (Enterprise Admins, User Store Admins, Domain Admins, Expert Contributors, and Domain Contributors), each of which can map to three “levels,” representing coarse-grained privilege and visibility scopes. You can define your own custom groups in the UI fairly quickly. Populating these groups with users, however, is another matter. You can assign individual users to groups easily enough, but doing this by hand becomes tedious. Hopefully, if you’re creating ad hoc groups, they’re small. LDAP support — including Active Directory — is available if you have an Enterprise license, which enables single sign-on and allows groups to be populated and synchronized from AD to the CMS.

In version 4.5, Enonic moved binary storage from database blobs to a disc-based system. It’s not a revolutionary change, but it is a sensible one in terms of performance. The Enterprise Edition has some ability to cluster instances of the product to support high-traffic sites with load balancing. Installation was also improved by using a “universal installer.” It is a GUI-based tool that deploys all the prerequisites and supposedly gets you up and running more quickly. The installer is based on Oracle’s “Java Web Start,” where applications can be initiated from a browser. As a result, out of the box, Enonic can be installed more quickly, and the correct JRE (Java Runtime Environment) is deployed automatically. To make use of this, you need to ensure that your environment has Internet access; some elements may be downloaded on the fly during this process.

System reporting still lags behind the WCXM marketplace. Enonic has a system audit log and a tracking tool that allows administrators to monitor performance, such as portal (back-end) traffic and front-end page and image requests. Version 4.6 wrapped these elements together in the administration console for easier access. For serious reporting however, you will still need to fall back on external reporting tools for data-driven audit logs.

Content

Contributor Experience

The Enonic CMS templating framework is based on XSLT. Any page displayed by Enonic CMS is generated from at least one XSLT style sheet. Enonic CMS’s strict dependence on XSLT technology is not unusual in this tier, but as a practical matter, it means that if you are creating anything more than a small number of relatively simple templates, you will need significant in-house domain expertise in XSLT programming in order to create, test, debug, and maintain the various templates. This is particularly true if you’ll be creating pages that dynamically pull data from the repository, or from other remote systems in your company. Note that Enonic does not bundle any XSLT development tools.

It’s tempting, but usually wrong, to think of XSLT template creation as a one-time “up front” effort in site creation. Unlike CSS or HTML, XSLT is a full programming language. Template development becomes software development. If your designs or layout will be complex or constantly evolving, you’ll need to allocate ongoing resources to XSLT development.

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

Version 4.4 added a set of example templates (that were extended in subsequent versions), which are skinnable in CSS (the vendor also provides three different skins.) Though this may not sound spectacular (many vendors offer example code), it may prove to be an accelerator for building your first Enonic site. The templates are quite good; however, they're not as high quality as Magnolia's template kit (Magnolia's kit will be sufficient for most simple web projects to stick to CSS design only; Enonic's really are just for starters).most simple web projects to stick to CSS design only; Enonic's really are just for starters).



Figure 264. Template library offers three choices of skins.

The system stores and manages not only site content but almost all configuration artifacts as XML (Figure 267). Developers can expect to spend some time editing raw markup in the browser-based UI. As with Oracle, Enonic CMS allows the administrator to create content types inside a browser-based interface, but the UI for doing so has the tired look of a Web 1.0 HTML form. There are no wizards or smart widgets to ease this task.

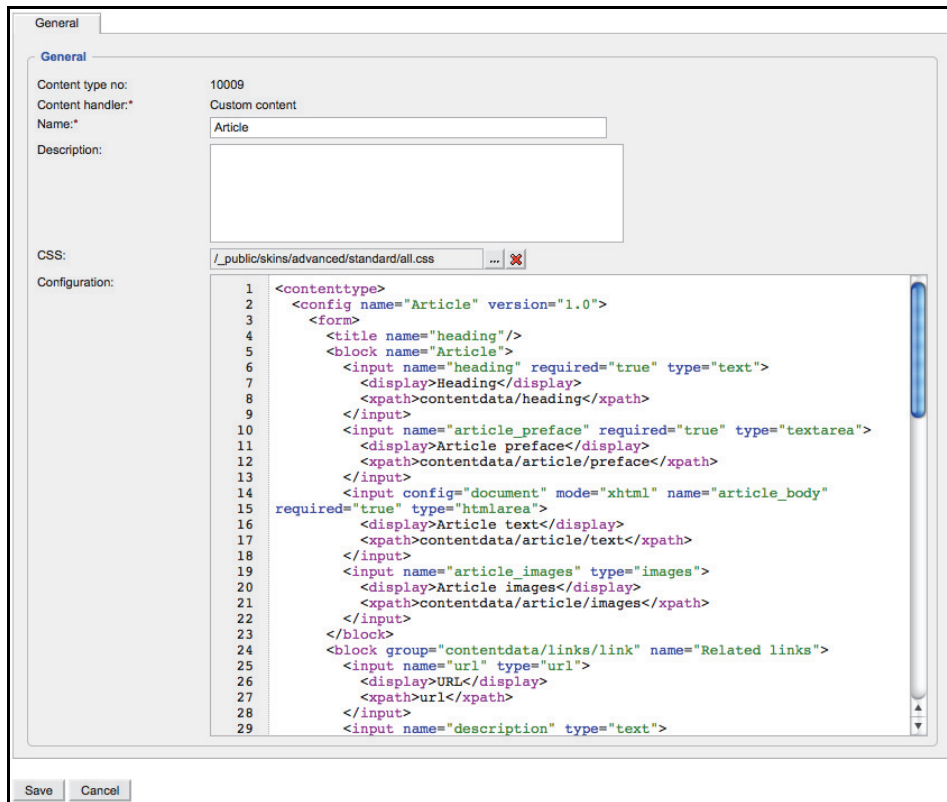


Figure 265. XML is used to define content types.

Contributing Content

Historically, Enonic CMS has had a somewhat bare-bones UI. Most UI features consisted of basic HTML forms, with none of the AJAX flash displayed in many other systems. Enonic is on the completely opposite end the spectrum of, say, Sitecore. Recent versions have incrementally added small conveniences to the Spartan user experience, but you still shouldn't expect a system that even tries to emulate the smoothness of a desktop application. As Enonic says, this interface "gets the job done" with a minimum of shine.

Though this may not "wow" you in a demo, it's not necessarily a bad thing: many other interfaces suffer from UIs that have bloated to the point that their "thin," web-based client is more intensive on the client PC than a fat client would have been. In some ways, Enonic can be refreshingly simple, though it will often leave you wanting for a bit more polish.

Out of the box, the interfaces are in English, Norwegian Swedish, and Danish; the company has been saying for some time it will be adding more languages, but no doubt will wait in doing so until a large customer demands it.

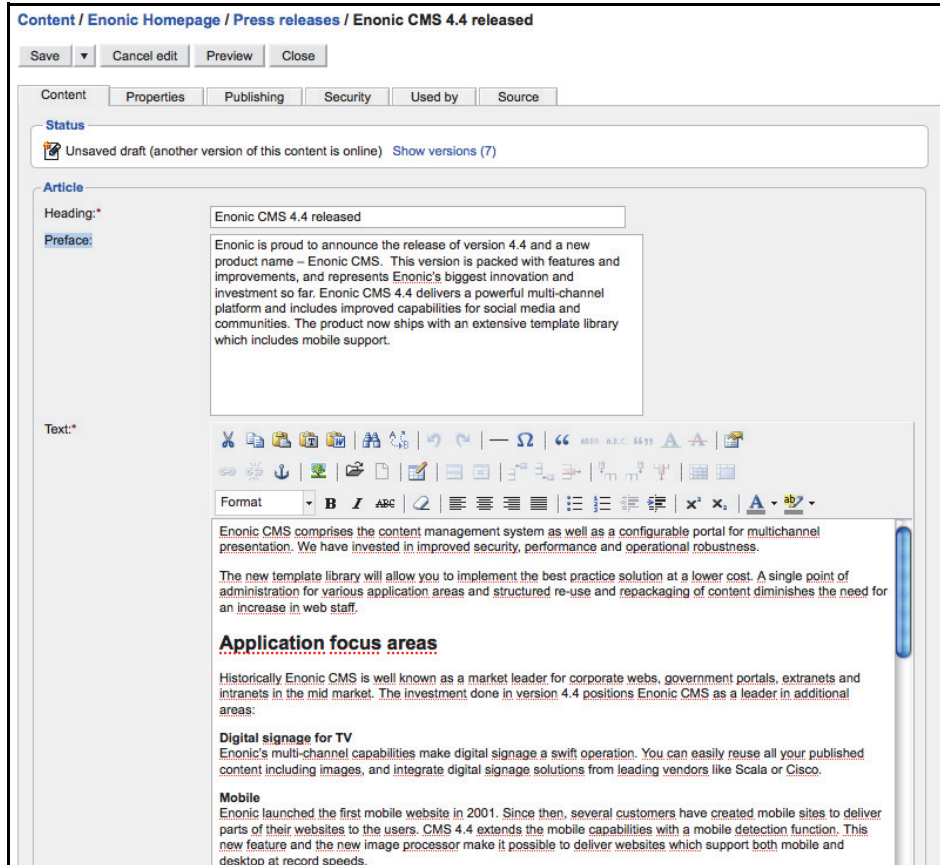


Figure 266. The authoring interface employs a functional — if uninspired — tabbed motif using TinyMCE under the covers.

Enonic CMS recently added an “In-Context Editing” (“ICE”) capability. While navigating a page in ICE mode, you’ll see a tool tip that updates as mouse over various pieces of content.

Lifecycle Services

In the past, Enonic workflow capabilities have been somewhat weak and this is something that has been addressed in recent versions (to a degree). It has a bare-bones approval system; items that are sent for approval can be queued for other users to work on, add comments, or approve. For some web publishing operations, this simplicity will suffice and in fact, may be ideal. Understand, however, that Enonic CMS has no concept of an automated, discrete, long-running process instance (a “workflow”), with parallel branches, joins, escalation, and so on, to automate the publishing process. Some improvements have been added in recent versions (adding draft status, for example) but it remains a weakness for Enonic. If you need such a system, look for another vendor.

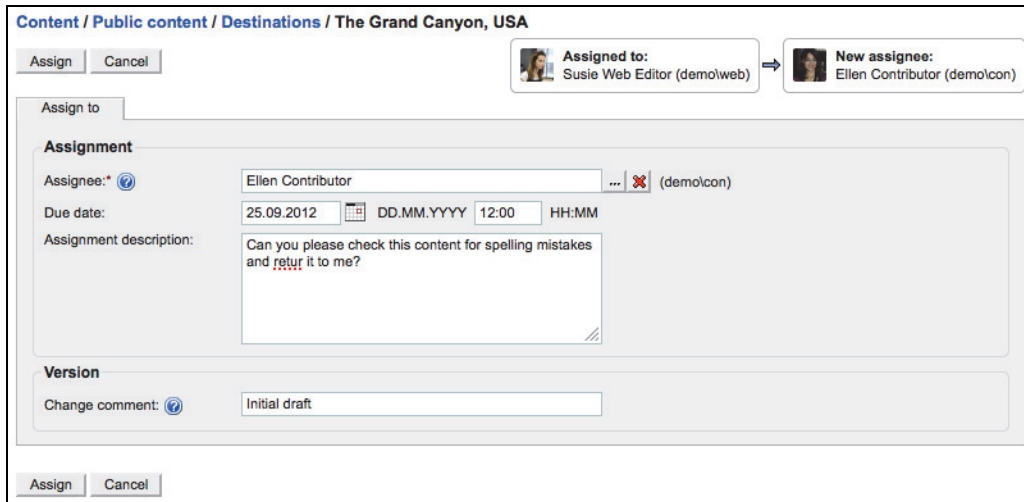


Figure 267. Workflow in Enonic is a very simple, assignment-based process.

Version control has improved. Multiple versions of content items are now retained and can be rolled forward and backward as required. A time-based “snapshot” of published content versions makes it possible to recall the precise state of any given page at any given moment of time. This is particularly necessary for users in regulated industries such as financial services.

Enonic’s support for globalization is not strong. Locales can be detected and redirected via Java Bundles (where a number of system resources are collected and then triggered by configuration scripts), but the process is highly technical and not manageable for business users. To make use of these features, you’ll need to keep a developer handy.

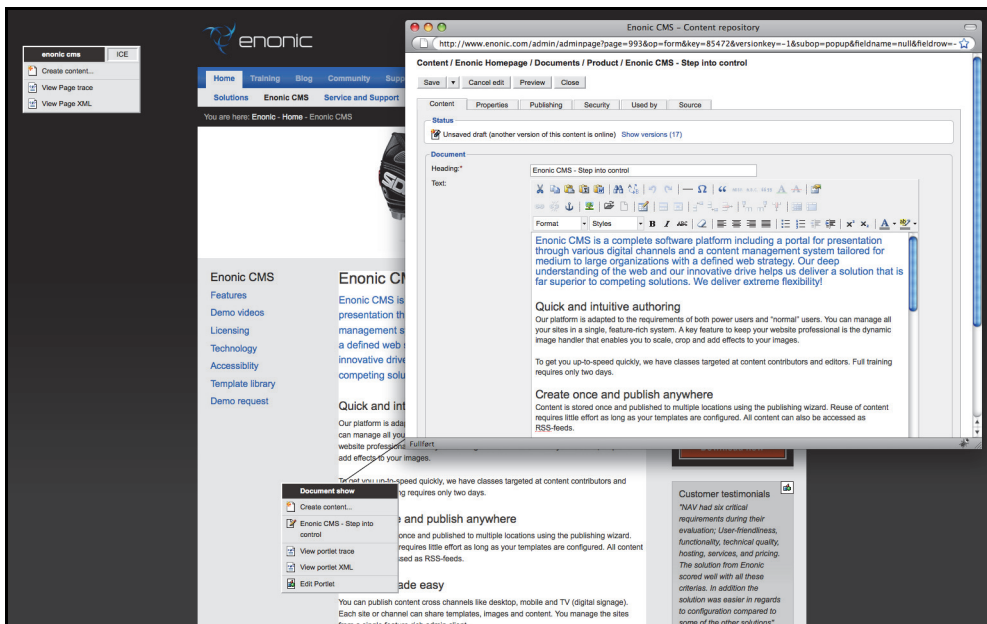


Figure 268. “ICE” is In-Context Editing; one tab is a structured-authoring form with a WYSIWYG editor:

Experience

Publishing

The URL structure in Enonic CMS is similar to what you might see in WordPress, in that it is based on the page title, which is helpful for SEO purposes. The generated URLs are prettified to form something like *http://hostname/en/folder/press-releases/press-release-title*. However, the actual URLs underneath the prettiness still retain the historic numeric URI in them: *http://hostname/folder/42143.cms*.

Unlike previous versions of Enonic, since version 4.5, this URL can be hidden, but it still acts as a “permalink,” even if the public URL is subsequently changed.

Version 4.4 added “device detection,” which allows you to serve content to various (mobile) devices from the same URL. Enonic CMS will identify the device a visitor is using, and display an appropriately reformatted page. In a sense, this is much more elegant than taking a “greatest common denominator” approach and publishing a separate channel for mobile. Likewise, most HTML and CSS designers have given up trying to address the ever-increasing chaos in providing support for various media stylesheets. Enonic CMS takes care of most of this for you. However, if you want to treat mobile as a separate channel — borrowing and reusing content, but not necessarily for the same website — this will be of little use.

Experience	
Publishing	
Standards Adherence	<input type="checkbox"/>
Multichannel	<input type="checkbox"/>
Mobile	<input type="checkbox"/>
Digital Marketing	
Site & Campaign Analytics	<input type="checkbox"/>
Testing & Optimization	<input type="checkbox"/>
Segmentation & Personalization	<input type="checkbox"/>
Social Media Integration	<input type="checkbox"/>
Promotional Campaigns	<input type="checkbox"/>
Community & UCG	<input type="checkbox"/>
Workplace	
Collaboration & Networking	<input type="checkbox"/>
Dashboard	<input type="checkbox"/>
Ancillary	
Site Search	<input type="checkbox"/>
Online Forms	<input type="checkbox"/>
Module Ecosystem	<input type="checkbox"/>

Digital Marketing

Enonic has not paid much attention to providing collaboration or community apps out of the box; don’t expect to find polished, ready-to-use blog or wiki functionality. You will find basic support for a handful of micro-applications (prebuilt servlets), including polls, discussion groups (forums), and a shopping cart. However, as a rule, the built-in capabilities need significant augmentation (i.e., custom development) to be of any value.

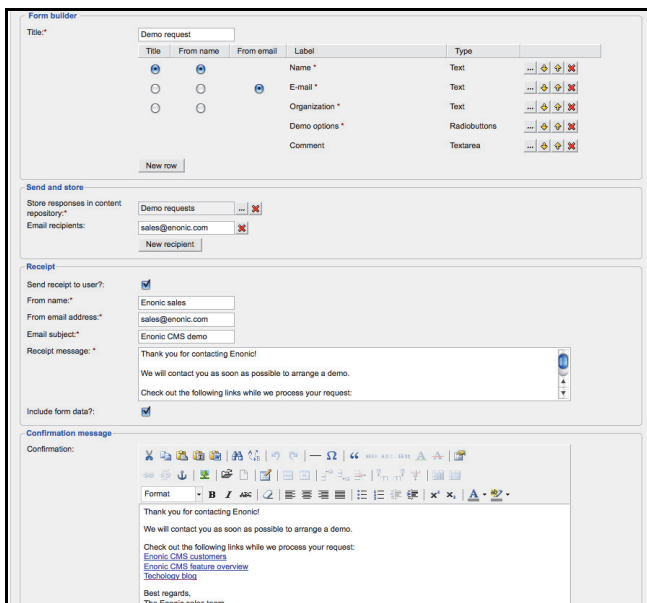


Figure 269. Formbuilder.

On the other hand, Enonic is quite capable in handling forms (which are treated like any form of content, and entered into the system).

Ancillary Services

There are no packaged visitor search facilities; customers turn to third-party providers.

Project Intangibles

Enonic CMS has two available packages:

- Enterprise – This has full functionality and no limitations in terms of users and websites. Starts at EUR 16K per node for a production server and EUR 8K per node for non-production environments like
- Community – This is the open source version under the AGPL 3.0 license with limited functionality and no official support. Note that the Enonic CMS API is licensed separately under Apache 2.0. Features like load balancing, LDAP support, and cluster diagnostics are not included in this edition.

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

These prices are relatively modest when compared to the regional competitors, and on larger deals, Enonic is known to be flexible in its licensing.

Documentation is somewhat improved, but remains comparatively cursory overall, and much of it is only available only in English. The company has started a community site, but it is still very light on traffic, and most threads are in Norwegian. Training is offered in Norway for customers and partners, but only in Norwegian. If you want training in English, you’ll need to make additional arrangements.

As with other regional vendors, you’ll find a significant depth of experience in the Norwegian home market, but beyond that, experienced consultants remain sparse.

Conclusion

Enonic remains a smallish Norwegian vendor on a vast global stage. They manage to remain relevant because of a strong web focus; clean architecture; developer-friendliness; and continuing adherence to standards (notwithstanding the termination of its prior JSR 168 support). Whether or not this combination is good enough to sustain the company remains unclear.

In favoring a clean, standards-driven architecture over client-side feature creep, Enonic has earned itself some flexibility. However, as competing vendors (e.g., CoreMedia, Oracle/FatWire, and Sitecore) continue to hone their user interfaces to a high gloss, Enonic CMS’s somewhat staid UI begins to feel outdated. Out of the box, it must be said that Enonic CMS’s feature set is neither impressively broad nor deep.

Still, Enonic CMS is built on solid foundations and appears to hit a sweet-spot for users who have modest content management needs, a modicum of in-house Java talent, and a limited budget. If that’s you, give Enonic CMS a look, but remember that this is a decidedly XSLT-oriented system that eventually will require some unavoidable, raw markup editing. Suitable skills are required.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

e-Spirit: FirstSpirit

www.e-spirit.com/en/

Vendor at a Glance

Specsheet	e-Spirit: FirstSpirit 5.0 Summary
Geography	Primarily Europe, some in North America
What's New	<ul style="list-style-type: none"> • New HTML5 browser client • Re-written thick Java client • Optional module for front-end content integrations
Strengths	<ul style="list-style-type: none"> • Simplified WYSIWYG editing, without unnecessary features cluttering the interface • Rich editorial functionality offered by Java application-based thick client • Virtual “time machine” allows you to roll back site content arbitrarily • Strong integration with SAP NetWeaver Portal, WebSphere Portal, Liferay, and Microsoft SharePoint • Facilitates integration of external structured sources (such as databases) into websites • Offers good workflow • Codebase is traditionally quite stable • Decoupled architecture will appeal to some enterprises
Weaknesses	<ul style="list-style-type: none"> • Browser-based thin client only can be used for content editing; for some other tasks, you need the thick client (applet) • WYSIWYG editing is not Word-like • Relatively inflexible site structures hinders more advanced globalization efforts • Lack of key audit trails and reports reduces enterprise value • Does not leverage mainstream Java standards — uses several arcane choices for some subsystems, including templating • Initial product implementation can become quite complex • Comparatively pricey for what you get • The product’s highly modular nature makes it hard to determine elements of the core platform
Potential Fit	Global Enterprise, Ultra-Large Single Site, Global/Enterprise Digital Workplace
Unlikely Fit	Informational Site, Microsites & Landing Pages
Compare To	CoreMedia, Adobe, Escenic, Magnolia
Operating Systems	IBM AIX, Linux, Microsoft Windows, Sun Solaris
Repository	Databases: IBM DB2, Microsoft SQL Server, MySQL, Oracle, PostgreSQL
Client	Any browser for content editors: Java client (Java WebStart) for template designer, administrators, and super users

Specsheet	e-Spirit: FirstSpirit 5.0 Summary
App Platform	Java container: Supports IBM WebSphere, Tomcat, and Jetty
Licensing	Starts at €80,000, for a base license that covers unlimited CPUs and servers. Additional feature modules are sold separately.
Ownership	Privately held, owned entirely by consulting company

Summary

e-Spirit was early out of the gate back in the late 1990s, with an object-oriented Web CMS based on Java. While novel at the time, today this approach will seem cumbersome to many customers, and the product’s technology has fallen out of the Java mainstream. On the upside, like German software giant SAP, e-Spirit maintains a steady, albeit slow release schedule — which is just how many large European enterprises like it.

e-Spirit offers strong integration with Enterprise Portals like SAP, IBM WebSphere Portal, Liferay, and Microsoft SharePoint, which renders it particularly more apt for intranet environments. If your company is headquartered in Germany, potentially considering any of these products, you should consider FirstSpirit for your basic web initiatives, where some of its features may help make your projects run more smoothly.

The product may not be a good fit for customers who need to build complex web applications that don’t employ a portal on the delivery tier.

Scenario Fits	
Simpler Site	
Informational	
Microsites & Landing	
Mid-Range	
Basic Digital Marketing	
Mobile Site	
Community Oriented	
Complex Site	
Advanced Marketing Portal	
Global Enterprise	
Multichannel Publishing	
Ultra-Large Single	
Digital Workplace	
SMB/Departmental	
Global/Enterprise	

Introduction

German CMS vendor e-Spirit was founded in 1999 by former employees from the internationally renowned Fraunhofer Research Institute. The company released FirstSpirit as a CMS, mainly focused at websites for German organizations. e-Spirit now has offices outside of Germany, and sells to international companies, as well as German companies with multinational presence. Their documentation is in both German and English. When compared to other European vendors, however, e-Spirit feels more like a regional player.

Today, the FirstSpirit division has 90 plus employees, mostly based at the Dortmund headquarters. Like many other European WCM vendors, e-Spirit recently established dual headquarters, with a Boston (USA) office opened in September 2011 to cover North America. Documentation is in both German and English. When compared to other European vendors, however, e-Spirit feels more like a regional player. More than 70 percent of its customer base is in Germany.

The company is known to be quite engineering driven, and in recent years, e-Spirit has formed important partnerships with IBM and SAP. e-Spirit’s productized integration to SAP Portal brings in some business, due to SAP’s enormous footprint in their German home market. Today roughly 30 percent of e-Spirit customers are integrating with SAP Portal, 10 percent integrate with WebSphere Portal and Commerce; 10 percent with Microsoft SharePoint, and an additional 5 percent with Liferay. The remaining customers use FirstSpirit as a traditional, baking-style CMS.

FirstSpirit version 4.2 was released in March 2009. The release added features to the Java client, and improved the smoothness of the web client operations, though for most tasks other than editing content, the Java client was still required. Version R4 was released in April 2011, which further smoothed out the usability of the web client, and added the first integration with external applications in the Java client. Version 5.0 was released October 2012, with a new HTML5-driven web client, which allows a greater range of capabilities within the “in-context editing” mode — without having to delve back into the Java client as often. The vendor also separated out administrative and advanced tasks from the web client into the Java client, thereby concentrating editorial activities (like content creation, content editing, image editing, publishing, etc.) within the web-based client.

Technology

Technical Administration and Security

FirstSpirit is written in Java, and a Java web container is required on the server. The system supports any certified Java servlet container, but most customers use either IBM WebSphere or the open source Apache Tomcat.

Interestingly, a database is not strictly required, as the FirstSpirit server uses Oracle Berkeley DB as an XML-based repository. However, it’s possible to use existing database installations for FirstSpirit data sources via a FirstSpirit internal abstraction layer based on JDBC. Most databases are supported, including commercial (Oracle, MS SQL Server, and DB2) and open source (MySQL, PostgreSQL, and Derby) alternatives.

Like many other systems, FirstSpirit uses a modular approach. The core covers basic functionality. You can license additional modules separately for multisite management, form creation, personalization, search, portal integration, and workflow. These all cost extra.

FirstSpirit is a baking system and publishes managed XML content via templates to static outputs such as XHTML pages. Like most baking systems, FirstSpirit can deploy unexecuted scripting code (such as PHP, ASPX, or JSP) to be executed by the delivery environment at request time. XML can also be transformed to other mediums, such as to PDF via XSL-FO templates and Apache FOP.

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

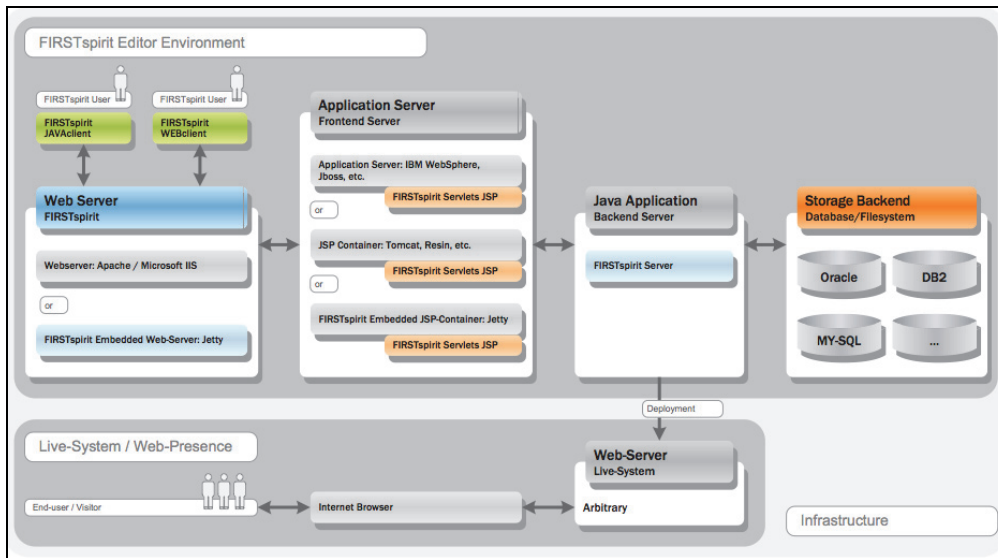


Figure 270. FirstSpirit is a Java-based application consisting of two tiers: A back-end server that contains core business logic, and a front-end server that communicates with FirstSpirit’s clients. Source: e-Spirit.

You can add users in FirstSpirit, or optionally, connect to LDAP or Windows Active Directory. Access control is defined by “permissions”; these are applied to a project (e-Spirit lingo for a site), or a branch of the project, and can be inherited down from there. The “permissions” are actually a set of attributes; the most basic right to an object is “visible,” ranging through change, create, remove, and modify permissions. You assign users or groups the specific permissions.

The product also has a “server monitor,” which tracks the technical status of your server. Note, however, that while the tool is very adept in informing you of how your databases, clusters, and deployment are faring, it doesn’t provide reporting on user activity. (There are no audit trails, for instance.)

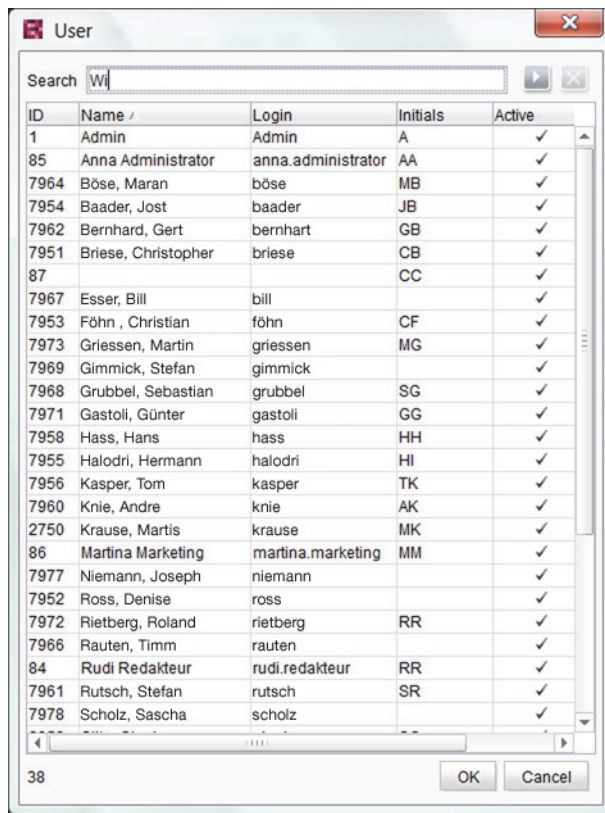


Figure 271. Managing users in FirstSpirit.

Development

e-Spirit suggests that what they provide is a framework, not specific functional elements, which means the system is flexible, but also effort and time consuming. If you're starting with the product, you'll have a large tabula rasa on your first project. Like its competitors, even though e-Spirit provides a sample project site that contains pre-developed sample templates and use cases, do not assume that this starter site will be all you need for your FirstSpirit implementation. Prospective customers should take note of this and budget/schedule developer training and education accordingly.

Developers can work with their IDE of choice, because the templates reside on the file system, but there is also the option to edit them through FirstSpirit's thick "JavaClient."

Templates in FirstSpirit are written in a proprietary XML-based

programming syntax (Figure 272) along with your XHTML and pure Java, and potentially BeanShell code. Although it is possible to write FirstSpirit module extensions in pure Java, which is the recommended approach by the vendor, we've seen this also done in BeanShell, which was used to extend functionality and allow developers to define functions that can be called from within a script. While BeanShell is a popular testing and debugging tool, its syntax can engender a comparatively steep learning curve, and FirstSpirit is the only product in this report to use it for templating or other logic. Other companies have chosen to use the ubiquitous JavaScript language with Mozilla's Rhino interpreter for this kind of scripting.

Alternatively, FirstSpirit also supports JSR 223 scripting languages like Python, Groovy, and Ruby for template scripting. While this significantly provides options for your choice of scripting languages, remember to test it out during the proof of concept.

```

Preview | Properties | Form | html (HTML) | print (HTML) | pdf (PDF) | webedit (HTML) | DTD
<CMS_PARAM name="case" value="1"/>
<CMS_CDATA_PARAM name="value"><![CDATA[<table cellspacing="0" ce
<tr>
<td align="left">${CMS_VALUE(set-st-currentPicture)}</td>
</tr>
</table>]]></CMS_CDATA_PARAM>
<CMS_PARAM name="case" value="6"/>
<CMS_CDATA_PARAM name="value"><![CDATA[<table cellspacing="0" ce
<tr>
<td align="center">${CMS_VALUE(set-st-currentPicture)}</td>
</tr>
</table>]]></CMS_CDATA_PARAM>
<CMS_PARAM name="case" value="7"/>
<CMS_CDATA_PARAM name="value"><![CDATA[<table cellspacing="0" ce
<tr>
<td align="right">${CMS_VALUE(set-st-currentPicture)}</td>
</tr>
</table>]]></CMS_CDATA_PARAM>
<CMS_PARAM name="case" value="2"/>
<CMS_CDATA_PARAM name="value"><![CDATA[<table cellspacing="0" ce
<tr>
<td width="100%" valign="top"><table cellspacing="0" cellp
<tr>
<td class="text">${CMS_VALUE(set-st-currentText)}</td>
</tr>
</table></td>
<td width="10">${CMS_VALUE(set-st-currentPicture)}</td>
</tr>
</table>]]></CMS_CDATA_PARAM>
<CMS_PARAM name="case" value="3"/>
<CMS_CDATA_PARAM name="value"><![CDATA[<table cellspacing="0" ce
<tr>
<td valign="top">${CMS_VALUE(set-st-currentPicture)}</td>
<td width="10"><table cellspacing="0" cellp

```

Figure 272. Note the product-specific variables used inside the HTML. Make sure to plan adequate training, and consider the impact of migration.

While FirstSpirit offers a rich API, their XML syntax seems outdated. It does not use a standard JSP tag library like the JSTL (Java Standard Tag Library) that most Java developers associate with the term “tag library.” Nor does it employ a commonly used templating framework, like Velocity or Freemarker; this seems somewhat strange for a Java-based system.

As with other baking-style systems, developers can add unexecuted scripting code (PHP, ASPX, JSP, et. al.) into presentation templates that are executed by the decoupled delivery tier.

e-Spirit calls content types “objects.” You define and create objects using a wizard (Figure 273); however, you will still need technical skills to make decisions on each element in the object. Note that you are not using Eclipse or any other standard IDE here.

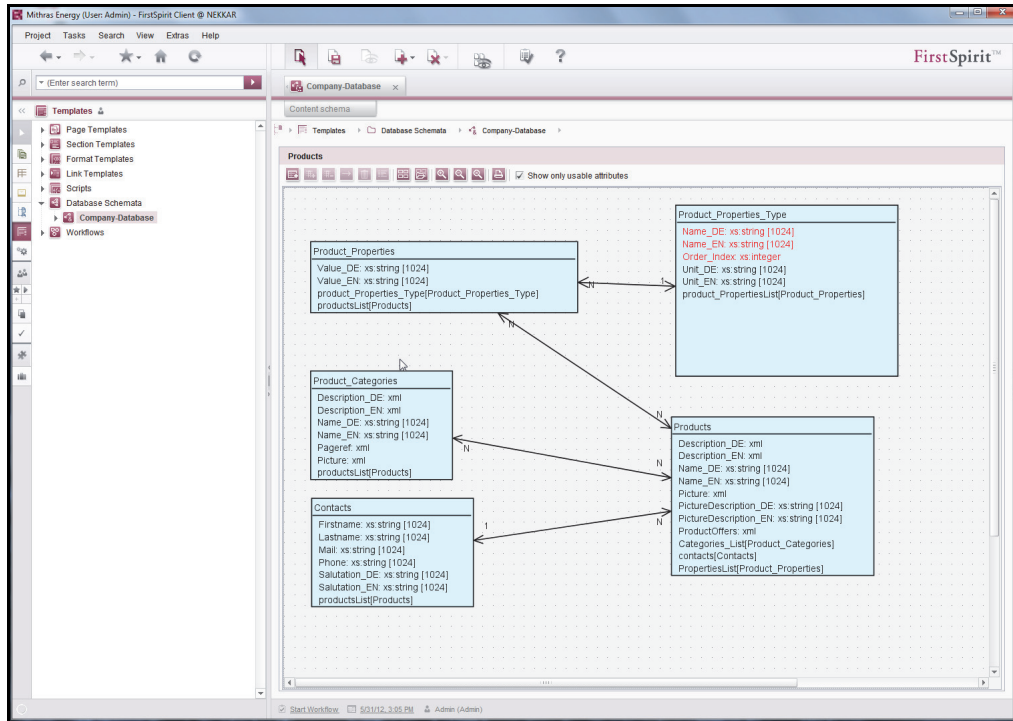


Figure 273. Mapping an external data source to the FirstSpirit CMS.

It's possible to aggregate data from external repositories. JavaClient (FirstSpirit's power-user interface) has a set of graphical user interfaces to browse, edit, and create external data sources. When you configure a new or pre-existing external data source in this way, you can configure FirstSpirit to create and manage history tables that provide versioning support for these data. This is useful for many FirstSpirit customers whose websites connect with ERP systems and other data-driven applications (and in practice, is also used to ingest large amounts of similarly structured content, such as news). There is wizard-style query builder that avoids the need to know SQL — although if a user has gotten this far into the user interface, chances are good that they already know SQL.

Performance

JavaClient requires a Java SE JRE 1.6/1.7, and needs at least 768 MB of RAM on the desktop, with Windows 7, Linux Desktop and Mac OS officially supported. By default, JavaClient talks to the server over standard HTTP, and encryption is handled by the application or the transport layer. Though JavaClient is started from (and can be installed from) the browser, it relies on the JRE available on a client machine. In a large enterprise, it can be hard to predict which JRE will be present, which can pose problems in running the JavaClient. It's also worth noting that for Windows 7, the 32-bit JRE must be used even on 64-bit versions of the OS, which might also cause some IT department headaches when constructing standardized builds.

For planning server-side hardware, e-Spirit offers a nice calculator, which will give you a good indication of how much RAM is required. It is possible to create a high-availability setup using FirstSpirit. This is accomplished using a cold standby server (it doesn't have to be in constant synchronization, like a hot standby) and a commercial high-availability solution, such as Veritas Cluster Service. This will then monitor the FirstSpirit server and ensure that it

continues running without problems on a different system, in the event of a software or hardware failure.

Content

Contributor Experience

Five different interfaces are available when working with FirstSpirit, although non-administrators likely will only touch the first two:

- A super-user interface called JavaClient, primarily for administrators, heavy users, and template designers. It requires a Java application to be downloaded. The Java client can be installed using a web-based installation using Java Web Start v1.2 or as a Java application.
- An editorial interface called WebClient is entirely browser based and is used by content contributors for the majority of editorial tasks like content creation and publishing. If you work with SAP Enterprise Portal, this is where your content contributors will primarily work.
- A Java-based administrator console called “Server and Project Configuration”
- A browser-based administrative console for manually monitoring FirstSpirit server called “Server Monitoring”
- Java Management Extension (JMX) based automatic monitoring console called “JMX Monitoring.” JMX monitoring can also be integrated into the enterprise monitoring tools that support JMX.

Both JavaClient and WebClient were overhauled in versions 4.x and again in version 5.0 to make editing a smoother task. However, you may find it still has some rather product-specific quirks. In general, customers seem to like the interface.

JavaClient’s UI extends across the width of wide-screen monitors, displaying a preview of the edited content at the right, highlighting the affected area. This kind of layout — a button bar, a directory tree pane, edit pane, and preview pane — is increasingly popular. (It is quite similar to CoreMedia 6’s Java client.) The preview — which in essence wraps a version of a local application, like a browser, into the Java client interface — also can be detached and displayed on a second screen. What is now the middle area — where the editing forms are displayed — can be “pinned” as a work area, enabling users to work with multiple tabs editing forms, and list views at the same time. Which application provides the previews can also be configured, so Firefox could be replaced with another supported web browser, such as Google Chrome. For other formatting previews, like mobile devices, you can plug in any emulator to perform the task. It’s flexible, but it’s another customization that you’ll have to support.

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

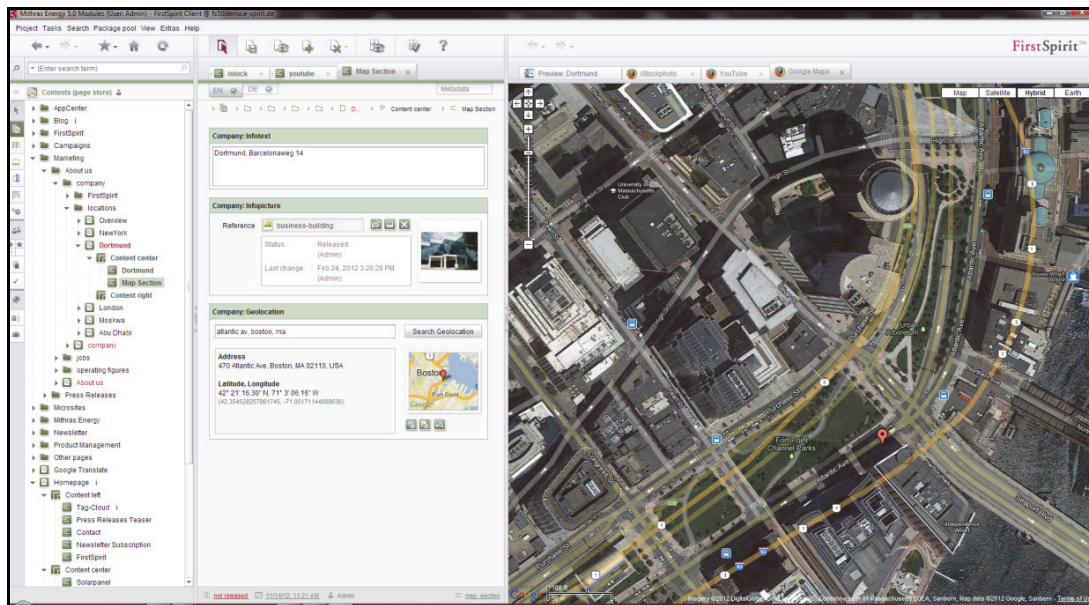


Figure 274. *e-Spirit’s AppCenter allows content to be populated from external applications, such as Google Maps.*

It is also possible to open Office documents in the preview pane, displaying the full Office interface. This works with Microsoft Office, Google Text, and (in beta form) OpenOffice. Although it may be useful for editors that frequently work with these source documents, any click outside the client is a potential distraction. Power users will beg you for large, HD resolution screens to fit all of it. FirstSpirit 4.2R4 introduced a similar kind of interface integration as the “AppCenter” for external Web Services, such as Flickr, YouTube, Google Maps, Salesforce, and Trados. Some of these are developed and supported by the vendor, while others are entirely your responsibility. This AppCenter allows assets to be dragged and dropped from these third-party services into page components (which e-Spirit calls “section templates”). To place a map or a video on a page, it’s quite a neat solution (albeit keeping in mind that you must build and support these specific integrations yourself).

Multinational enterprises should note that the interface is available in German, English, French, Spanish, and Russian. The English translation is somewhat rough and could use some improvement. It is possible to translate various text strings in the UI using standard Java resource bundle files, although FirstSpirit offers no simple interface for this.

Contributing Content

The WebClient provides an in-context editorial interface with menu options inserted into rendered pages. Unlike other products that offer in-context editing, by default, FirstSpirit’s buttons do not collapse entirely, so they tend to distort the page layout (though much less so than in previous versions of the product — Figure 275). To modify a content item, the editor clicks on the *edit* button, which displays an in-line edit dialog. Though not as slick as other editors (where you edit the text on the page, displayed in the layout and style of the page), this has the advantage of showing the full form, while not jumping to a separate edit screen. It includes a spell checker for which there are many different (freely available) language dictionaries.

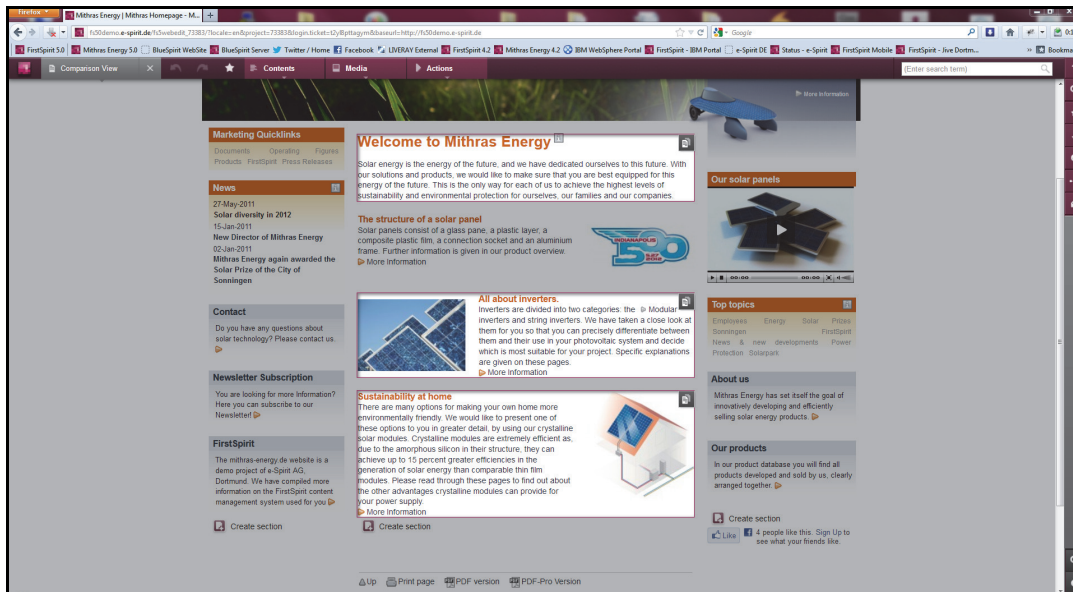


Figure 275. The 5.0 WebClient still distorts the page layout, but is much less obtrusive than in previous versions.

FirstSpirit uses its own rich text editor, because rich text content is stored in XML. Your contributors will notice that FirstSpirit's rich text area is not entirely WYSIWYG. Like some XML editors, styling directives are called out in a presentation-neutral way, in the form of annotations. The argument for this strategy was that the presentation templates for different publishing channels would interpret these directives differently at page render time; however, with the editing displayed in the page itself, users may find themselves toggling the edit mode on and off to see the actual result.

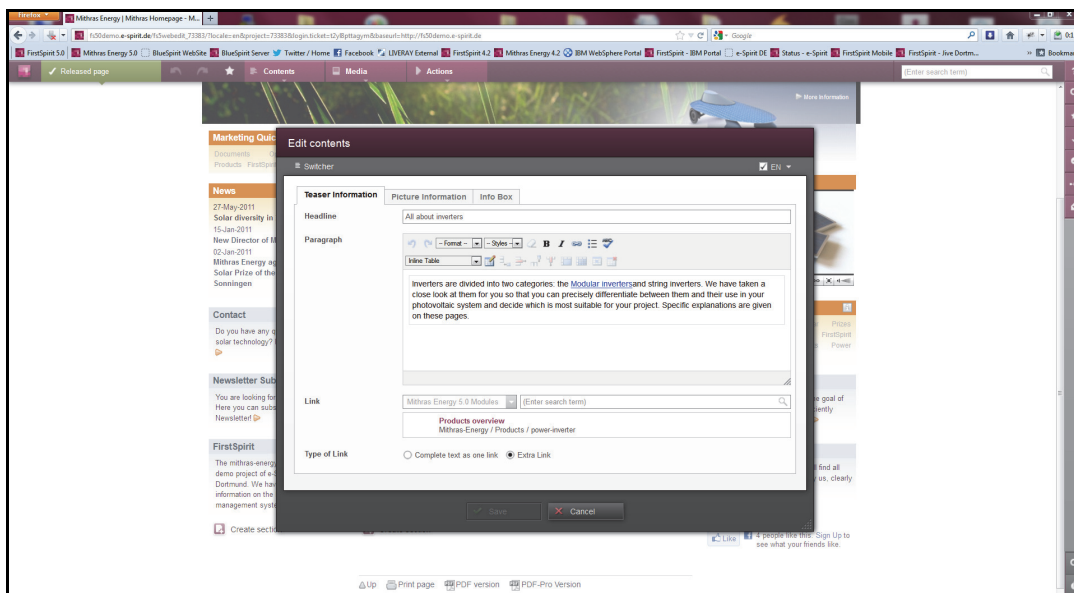


Figure 276. e-Spirit employs its own proprietary editor for in-context editing; it's simple, but functional for basic editing tasks.

The more powerful JavaClient has a tree-based navigation to browse pages and the content sections within each page. The JavaClient exposes FirstSpirit’s content reuse functionality. By dragging and dropping content from one location to another, a user can either copy or create a read-only reference to the original. FirstSpirit supports dependency management, so that a user is warned when he tries to delete a referenced asset. Similarly, links authored in the rich text editor are managed and registered with the dependency management capabilities.

JavaClient has a nice interface for editing images. This allows you to do basic manipulation, such as cropping and resizing. While by no means a replacement for the likes of Photoshop, this is convenient for a quick slice and dice. Images and other media are stored in a dedicated repository (the “Media Store”), and these crops are stored as separate physical assets, with smaller assets as database “blobs,” and larger ones in a file system. Although this is faster than the alternative on-demand transcoding (where just the crop data is stored), if you plan to have a large number of standard re-sizes or crops, you’ll need to plan your storage requirements carefully.

Using AppCenter, frequently used applications (like online image and video databases, web and social media analytics, and email marketing services) are available to editors within the Java client. The content can be imported or dragged and dropped into your site or content repository. e-Spirit claims that this integration can be achieved with just a couple of days of development effort. Customers tell us that this is achieved via DOM-based web clipping, so test this carefully against your needs. If it works for you, AppCenter integration should improve editorial productivity on complex sites and intranets.

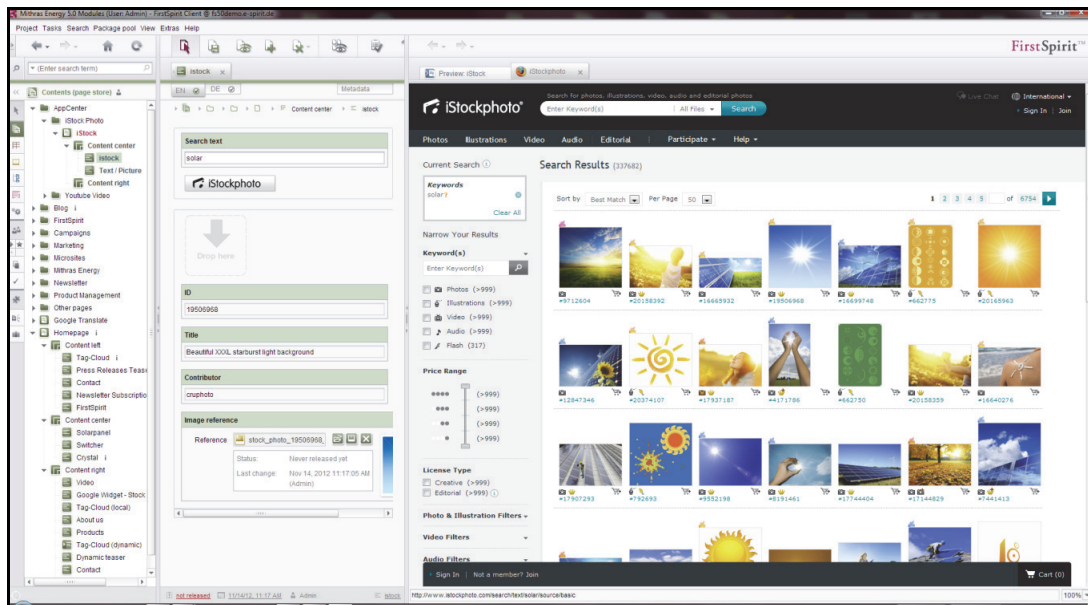


Figure 277. An example of an AppCenter search integration using iStockphoto.

Content Lifecycle

Workflow is offered with a separately licensed module. Simple or complex, various approval processes can be created using the graphical workflow builder. However, automated tasks require programming additional modules in Java. Workflows, once defined, can be allocated to any number of objects within FirstSpirit, and this allocation applies for all objects in the

respective sub-tree. This means that workflows can be applied to your content, structured elements, media, and templates. In addition, rights defined in the workflow can be modified for each object's assignments. For example, if an existing workflow describes the creation of press releases (including approvals and publication), then this workflow also can be used for the creation of news articles — even if other people or groups are responsible. Workflow status is communicated in the WebClient and JavaClient interfaces through a color-coding scheme.

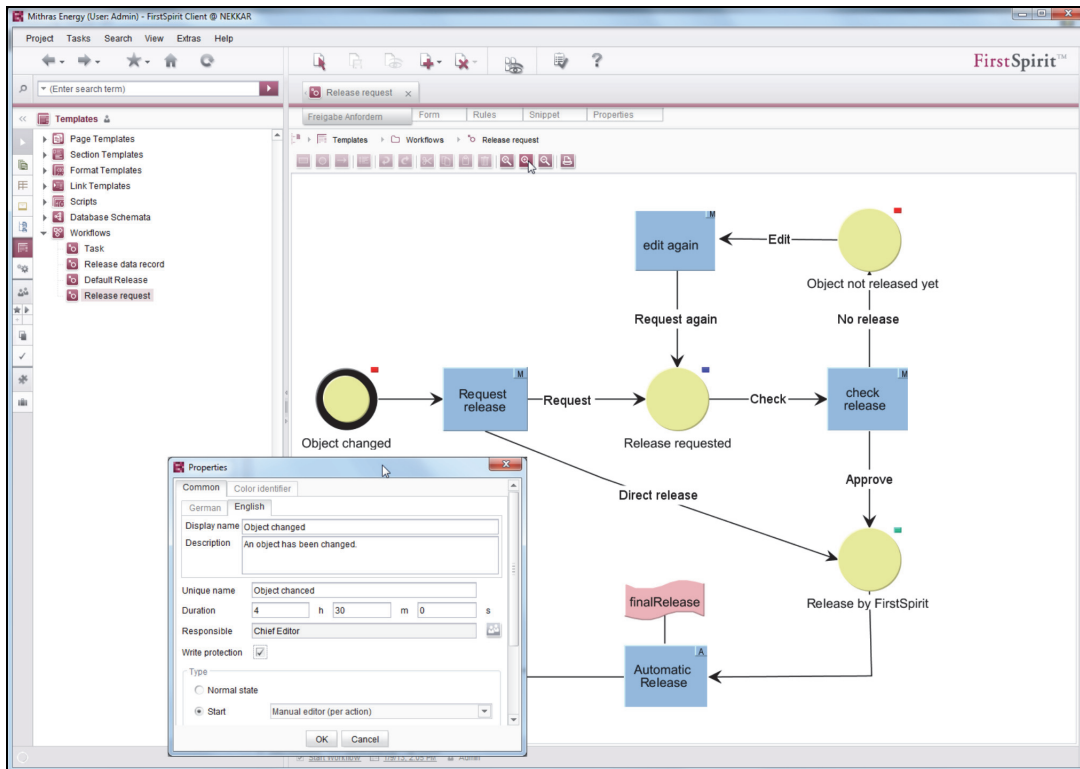


Figure 278. You arrange your workflow visually using the JavaClient. Individual status is symbolized as a circle, activities as rectangles, transitions as arrows, and scripts as flags. An “A” in the upper corner of an activity signifies an automated task containing script code.

As a product, FirstSpirit is centered on the idea of managing “projects,” which can include one or more sites. This gives you the benefit of sharing content, as well as user privileges across sites, as long as they reside in the same project. There is an optional module to share content, media, and templates across projects. Note that unlike systems such as SDL, it’s not easy to share or inherit and then change structure across sites. The system manages content in a hierarchical site tree. Each page contains sections that are essentially rich text blocks. A single complete section can be reused across multiple pages, where the original and the references are read-only. Alternatively, a section can be copied into another page, creating a new, independently editable instance of that section.

In the spirit of allowing for easier component-level content sharing, FirstSpirit aids its page-centric approach with an additional object-centric content model, where content editors can create or edit an “object” (e.g., news article), which is then placed on the defined page(s) on

the website as driven by the logic programmed into the templates. This object-centric content model also applies to translations of content components.

Translations of content are bound to the same object (and displayed as tabs on the editing window). The advantage is that it will be immediately apparent whether the content has been translated, and it is easy to see what has to change in corresponding languages if a change is made. This also allows for the relatively rare support of both language-dependent filenames (i.e., “danger.jpg” or “peligro.jpg” for the image of a warning sign) and language-dependent versions of the object (i.e., an image of the warning sign in English or in Spanish).

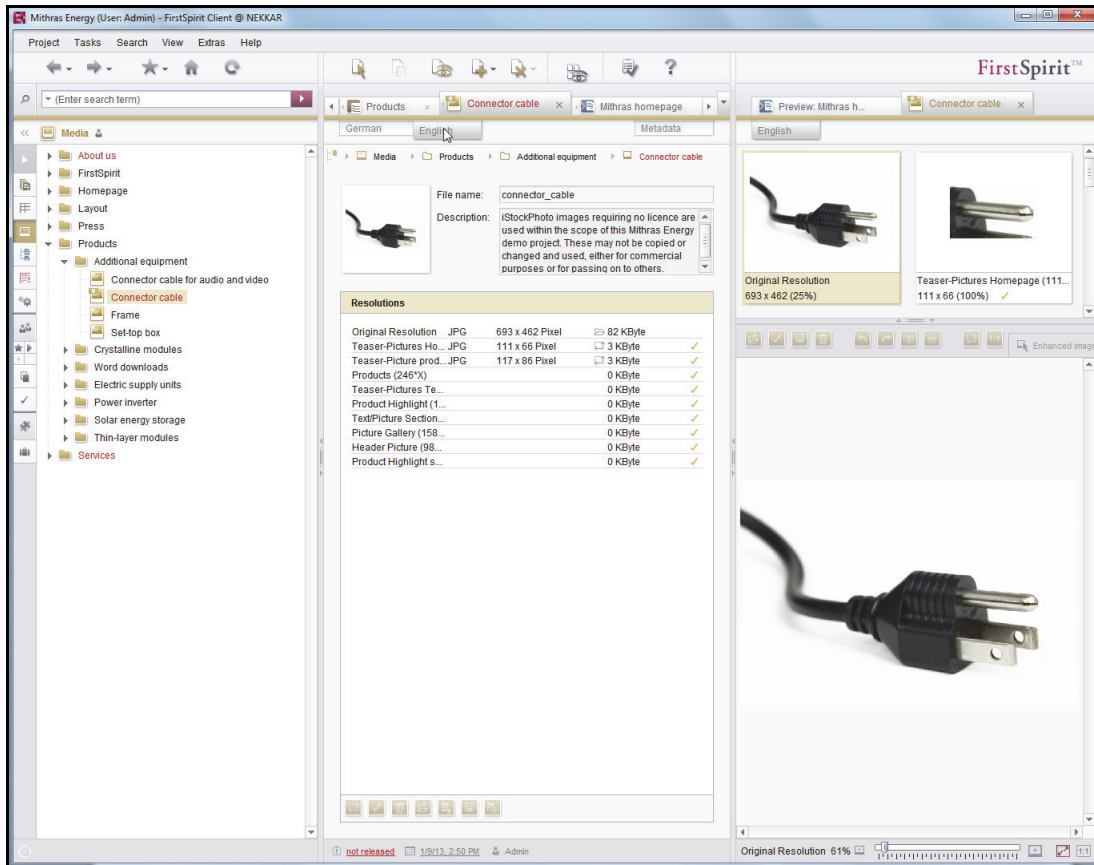


Figure 279. FirstSpirit supports multiple versions of the same content object for different languages and resolutions, including a language-dependent filename.

This works very well for common content and, for instance, product databases, but is far less flexible when there are loose dependencies between country or language sites. For example, if you were marketing the city of Hamburg to German tourists, they would be aware where Hamburg is within the country; to potential visitors from the US, you would have to explain this carefully. FirstSpirit is not well suited to the flexible content reuse in the different sites (with relatively unconstrained ties between language versions) required in such scenarios.

FirstSpirit supports what it calls “historization,” which essentially is a roll-back service that allows you to generate a temporary snapshot of your project or an individual page as it was at a specific time in the past. Note that as a consequence, content in FirstSpirit is never completely deleted or overwritten; this may be inconvenient for enterprises that are more

sophisticated about records management and discovery liability, and wish to permanently dispose web content, or simply move older object versions to an external repository or backup system.

Experience

Publishing

Many FirstSpirit customers deliver content through a portal. e-Spirit provides integration with SAP NetWeaver, WebSphere Portal, and Microsoft SharePoint. Thirty percent of FirstSpirit customers integrate with SAP portal. Twenty five percent integrate with WebSphere, SharePoint, or Liferay. Given SAP’s strong presence in Germany, e-Spirit is winning more and more licensees who already deploy SAP Portal. In this case, FirstSpirit handles the management of content and then publishes this to files (HTML, XML, or JSP) for SAP Portal to deliver. This same integration approach applies to IBM WebSphere and SharePoint.

Fellow German competitor CoreMedia also provides an SAP NetWeaver integration. Instead of managing the content in the CMS and pushing it out to NetWeaver (as FirstSpirit does), CoreMedia employs SAP “iView” portlets to show editing and publication from the CoreMedia CMS. If integration with SAP’s portal is essential to you, you will want to compare the two vendors’ differing approaches carefully.

Experience	
Publishing	
Standards Adherence	
Multichannel	
Mobile	
Digital Marketing	
Site & Campaign Analytics	
Testing & Optimization	
Segmentation & Personalization	
Social Media Integration	
Promotional Campaigns	
Community & UCG	
Workplace	
Collaboration & Networking	
Dashboard	
Ancillary	
Site Search	
Online Forms	
Module Ecosystem	

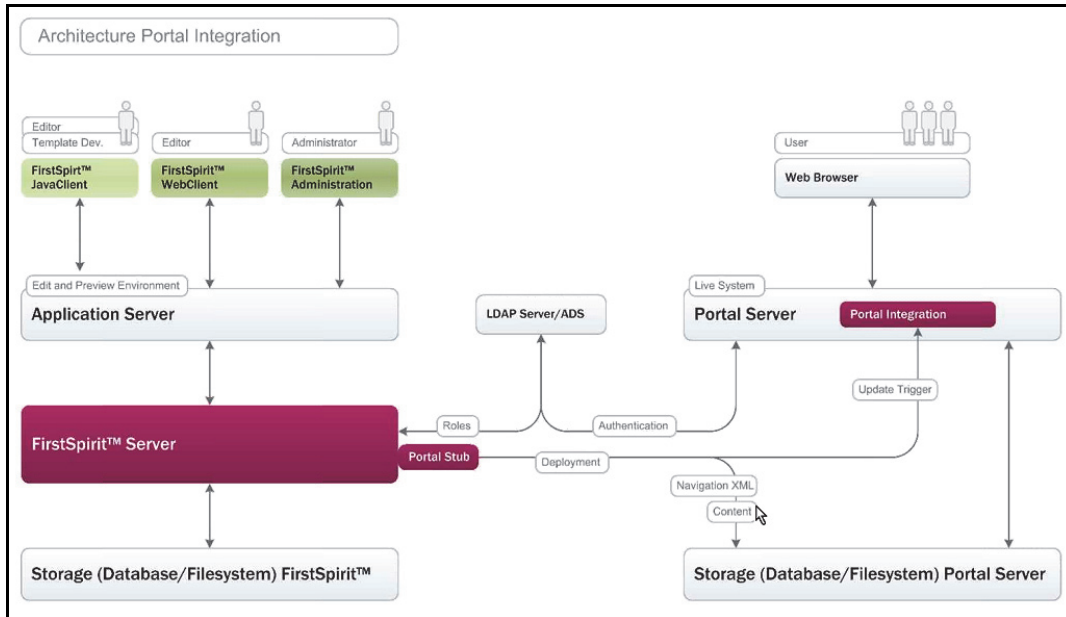


Figure 280. FirstSpirit's portal integration architecture.

The fact that FirstSpirit does not integrate with these portal products through a standard (like the JSR 168 Portlet spec) has advantages and disadvantages. By delivering content as native portal objects (and not using a portlet abstraction), FirstSpirit achieves tighter integration and leverages more portal functionality. On the downside, this means that e-Spirit can only support a limited number of portal products. Examples of FirstSpirit's integration include the ability to use the portal's search engine to find content, and the ability to assign portal roles to content, so that the portal can apply access control and personalization logic at request time. FirstSpirit can also create navigational elements that the portal transforms into portal links. By contrast, CoreMedia's content will be relatively disparate from the portal; it will not "know" what content is displayed, only that it's a CoreMedia block. Additionally, it will be difficult to integrate search and navigation.

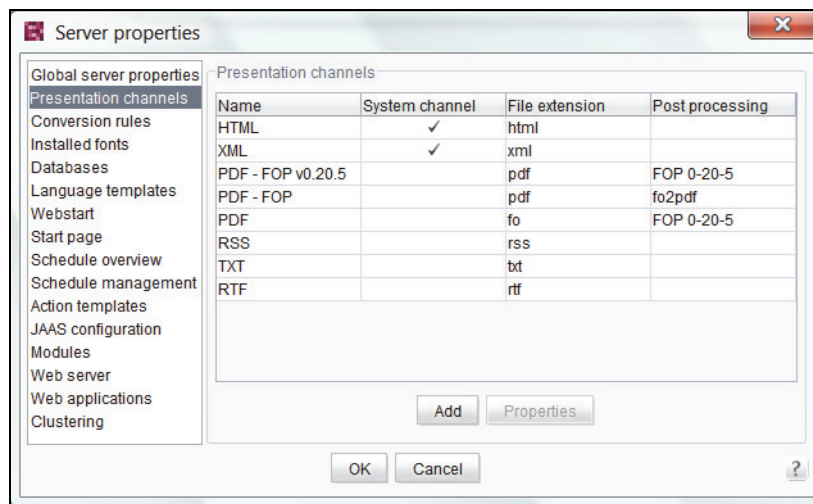


Figure 281. Managing presentation channels in FirstSpirit.

From a multichannel perspective, the software can push content to various channels; note however, that this is not particularly flexible — and creating channels with different content from the same FirstSpirit source can be quite difficult.

Digital Marketing

FirstSpirit is comparatively light on digital marketing services, with only basic website personalization functionality. Many customers use portal technology for these services or add them at code level within the published JSP pages. FirstSpirit states that it primarily provides a “framework” on which development and an increasing number of small, functional modules can be added. Some of these modules may add light digital marketing facilities, such as social publishing and basic email marketing.

Ancillary Services

Site search is available via a separately licensed module. With this module, content is categorized and the results are filtered, based on user rights as defined by FirstSpirit. Note that the additional module for personalization is also required, to ensure that protected information cannot be found using the search module.

Search is available in the both the web client and Java applet — depending on user privileges. You should note, however, that the search engine is not built to index content from outside of the repository.

Vendor Intangibles

With respect to services and integration, the partner model is mixed. Projects can be implemented with a partner or directly with the vendor. It is worth remembering that most partners receive a commission based on the number of licenses they issue (i.e., they are resellers as well as consultants). Because FirstSpirit is owned by a 750 employee-strong consulting company,

working with e-Spirit directly may be the least expensive option for licensees, providing that e-Spirit adequately can staff your project. According to licensees, working directly with e-Spirit also gives you the advantage of direct access to their engineering team.

A few partners mention that they like working with e-Spirit, but also that they struggle to convince larger, more global German customers to use FirstSpirit, since e-Spirit is less internationally oriented and has a comparatively small footprint. e-Spirit vehemently disagrees with this characterization, and points to multinational customers supporting global implementations and its new North American office.

Customers generally report that they are happy with support, and the company has effectively rolled out bug fixes on previous releases, which means that some of the usual pressure to upgrade has been alleviated.

FirstSpirit keeps a relatively slow release schedule. According to the company, this is indicative of the thoroughness of development; it is a similar pace as SAP — reassuringly familiar to many e-Spirit customers. The main releases in 2010 and 2011 were 4.2R2 and R4, both containing mostly minor fixes, not big updates. The company had promised to release

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

version 5 at the end of 2011, but it wasn't released until Q4 2012. Our impression is that e-Spirit seems focused on the growing SAP market and enhancing the product where SAP is weak. This means easier entry of content and lightweight document management (like Microsoft SharePoint). However, the company also reports several large wins where the SAP integration wasn't a consideration.

Rather than licensing by implementation scale (i.e., the number of CPUs), e-Spirit licenses its offerings by feature. There is a basic license for just getting started with the product (license per legal unit, e.g., GmbH, Ltd., Inc.). You then pay additionally, depending on how many channels you need to serve (intranet, Internet, extranet). Additional modules are extra, and there is an added rate for each additional concurrent user. Total license price for FirstSpirit starts at €80,000, with a typical deal being around €150,000. On the plus side, this model does not discourage customers from supporting sufficient development, QA, staging, and production environments. Unfortunately however, it is not always easy to determine which modules are worth buying.

Conclusion

Since the company began back in 1999, e-Spirit has taken a scientific and academic approach to running their business. They've been patiently financed and staffed mainly by former colleagues from the Fraunhofer Research Institute. Their scientific approach clearly resonates throughout the systems, and seems to play well in their German home market.

Don't be fooled into thinking that those slick demonstrations of JavaClient and the AppCenter integrations are out-of-the-box features; they're something that must be developed on top of the framework. With the current architecture, it will be challenging to expand outside of Germany, where seasoned competitors tend to use modern approaches to templating, authoring, and customization. On the other hand, you may value e-Spirit's integration with enterprise portals, particularly SAP Portal. If you like the company's Java and web clients, the product is one of few with a stable codebase.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

eZ Systems: eZ Publish Platform

ez.no

Project at a Glance

Specsheet	eZ Systems: eZ Publish 5.0 Platform Summary
Geography	Global, with a major footprint in Western Europe
What's New	<ul style="list-style-type: none"> • Major architectural overhaul of the CMS • Embedded Symfony 2 web application framework as product's underpinnings • Rewrite of the public (PHP) API and REST-ful APIs
Strengths	<ul style="list-style-type: none"> • Simple to install • Templating is comparatively easy to learn due to more adherence to a standards-compliant industry framework • Flexible and extensible development framework • Strong support for multimedia in general, and video in particular • eZ Find extension for internal search engine is based on Lucene/Solr • Comparatively easy to use in-context editing interface • Established, growing, and supportive community • Backed by a commercial company and VC funding
Weaknesses	<ul style="list-style-type: none"> • Local translations vary in quality, and globalization functionality remains incomplete • Scalability and performance issues traditionally haunt larger eZ implementations • Limited eZ Find search utility adds Java to PHP-based environment • Documentation (while improved) can be spotty • Administrative UI is difficult to use • Scarcity of tangible resources and support in North America • Continued turbulent internal changes on the executive levels • Online marketing capabilities are brand new and lack maturity
Potential Fit	Basic Digital Marketing, Informational Site, Multichannel Publishing
Unlikely Fit	Microsites & Landing Pages, Community-Oriented Site, Global/Enterprise Digital Workplace
Compare To	Ektron, TYPO3, Drupal, Plone, EPiServer
Operating Systems	Windows, Linux, Macintosh, Solaris, FreeBSD, HP-UX
Repository	Databases: Microsoft SQL, MySQL (on demand only), PostgreSQL, Oracle
Client	Browser: Chrome, Safari, Firefox, IE, Opera
App Platform	PHP5
Licensing	Dual: Free GPL or proprietary licenses, average deal size US\$40K
Ownership	Privately held

Summary

“eZ Publish Platform” is a Norwegian-based dual-license (commercial and open source) content management system and development framework, based on the popular LAMP platform.

The project has a large and generally helpful community, which has grown rapidly during the last couple of years. Backed by outside funding and a comparatively large number of employees, the privately held corporate parent, eZ Systems, serves as a community director and offers product support and professional services.

Consider eZ Publish Platform in particular for Basic Digital Marketing scenarios, simpler Informational sites, or Microsites.

Over the past few years, eZ Publish Platform has sharpened its focus on the media and publishing industry, where it has successful implementations with companies like Lagardère (publisher of Elle magazine, Paris Match, and Premiere), Prisma Presse, Orange, EMI, Wall Street Journal Asia, and The Christian Science Monitor. However, some of the very features that make eZ Publish Platform better suited for media and publishing also render it less well fit for traditional enterprise scenarios. This is especially true for Digital Workplace scenarios, where you might like better document handling and collaboration features.

Scenario Fits	
Simpler Site	
Informational	<input checked="" type="checkbox"/>
Microsites & Landing	<input type="checkbox"/>
Mid-Range	
Basic Digital Marketing	<input checked="" type="checkbox"/>
Mobile Site	<input checked="" type="checkbox"/>
Community Oriented	<input checked="" type="checkbox"/>
Complex Site	
Advanced Marketing Portal	<input type="checkbox"/>
Global Enterprise	<input checked="" type="checkbox"/>
Multichannel Publishing	<input checked="" type="checkbox"/>
Ultra-Large Single	<input checked="" type="checkbox"/>
Digital Workplace	
SMB/Departmental	<input checked="" type="checkbox"/>
Global/Enterprise	<input checked="" type="checkbox"/>

Introduction

The project started in Norway in 1999, when the eZ Systems founders got together to work on web publishing applications. In 2000, the company was registered, and a few months later, it had secured its first round of financing. Before the end of 2000, eZ publish was released as open source and became the main focus of the new company.

Version 5.0 was released in November 2012 and marks a substantial overhaul of the product, now named “Platform.” A new set of APIs was added along with support for the Symfony web application framework as well as the integrated Twig templating engine to improve developer experience. Although the legacy APIs and proprietary templates are still supported for backward compatibility, using any of the new functions requires knowledge of other pieces in that stack.

Another major trend in eZ’s strategy has been a push for commercialization and it has courted enterprise customers with its addition of 24/7 support SLAs and general price increases for enterprise subscribers — with a minimal entry point at US\$15,000.

As of this writing, eZ Systems had 105 employees with its operational headquarters in Cologne, Germany. Other offices include Norway, Denmark, France, Germany, Italy, Singapore, China, the UK, the US, and Japan. Experienced consulting partners are easier to find across Europe, whereas in North America they’re emerging at a slower pace.

Entry into North America has been slow and halting. A Vancouver, Canada, office closed in 2008, but by the end of that year, CTO and co-founder of eZ Systems Bård Farstad moved to Chicago to try a second push into the region. Farstad soon went back to Norway, and currently the US office is based in New York with two employees. Make no mistake: Apart from New York, eZ currently has very limited resources (down to 1.5-2 SE/technical resources) between Boston, New York, and various changing locations in the US. Thus, support still comes from a single center in Lisbon, Portugal, which for most is in another time zone.

This can become a major issue, and as one US customer notes, “We have to adjust to eZ’s time zone (CET) to get sufficient support.” To be fair, support quality has not been questionable, eZ is responsive, and “it doesn’t feel like open source,” the customer concluded.

Even though the company has reported some successes in the US, signing up Clear Channel and the Christian Science Monitor, overall the entry has been rocky, with only a handful of consulting firms acting in the implementer and system integrator capacity. Finding qualified eZ developers may be difficult.

In 2012, eZ Systems (yet again) changed its positioning to focus on “customer experience management.” However, there’s little on the product side to support the claim that eZ is “a global, feature-complete CXM platform.” Therefore, you should take this message only as the vendor’s aspiration at this point; CXM is a discipline and a strategy — not a set of technologies.

eZ Systems has described its offering over the years as an “Enterprise Content Management” system, and then as “Enterprise Open Source Web Content Management.” Prospective customers will find eZ Publish Platform is decidedly a web content management system that competes directly with TYPO3, Drupal, and other PHP-based systems.

Technology

Technical Administration and Security

As with many other open source content management systems, eZ Publish Platform is based on the popular LAMP platform. This means that the entire system runs natively on Linux, with the Apache webserver, MySQL as the database, and is written in PHP. Although eZ technically supports multiple databases, it runs best on MySQL 5.1 or higher. You can also run it on Windows and IIS, but doing so would put you in the minority of the PHP community.

Starting in version 4.0, eZ Publish Platform re-architected the application, using its “eZ Component” framework to take advantage of PHP5 (more about that later). In version 5, many of those eZ Components are being deprecated. Those that stay supported — like eZ Cache, for instance — are up to eZ’s discretion. Therefore going forward, you may not be able to use your custom-developed components you created in the past and will have to rewrite and/or switch to other approaches for application development and customization.

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

In version 5.0, eZ completely overhauled the underlying architecture, basing it on two entirely new eZ developer notions: the Symfony application framework and the Twig templating engine. Certainly some developers were familiar with these technologies prior to version 5, but the entirely new architecture will require some getting used to from a technical point of view.

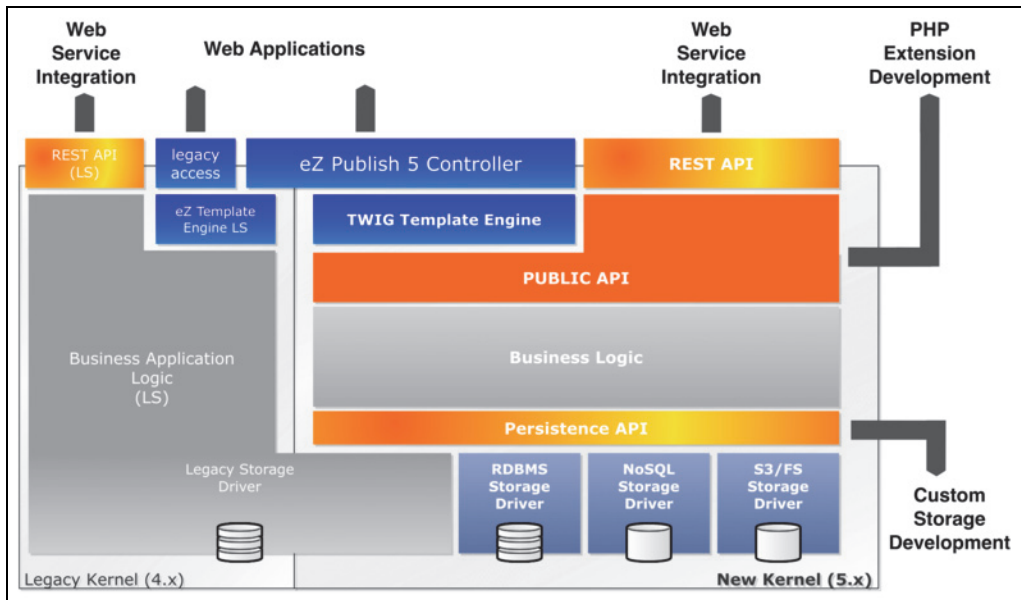


Figure 282. The eZ Publish Platform CMS architecture as of version 5.0 released has an overhead of legacy and new components. eZ calls it a “duality,” but you may call it “an existential headache.” Image source: eZ Systems.

Alongside the existing REST API (introduced in version 4.5), templating engine and storage driver, there is now an entirely new set of APIs, which allow for additional storage options via a Persistence API — including the plugging in of NoSQL repositories — as well as a new/replacement REST API for Web Service integrations, and a public API for PHP extension development.

All in all, backward compatibility for version 5 will remain an issue for quite some time and if you are an existing client, this will affect your old templates and modules, among other pieces of code. While eZ has created ways to support old code — such as enveloping old templating code in the new Twig templating language — you will want to refactor all of your templates and clean them up once you move completely onto Twig templating.

If you are a new eZ customer and want to use any of the new features, take advantage of the new APIs; you will have to learn the new templating and application framework. Because Symfony is a fairly well-regarded framework in PHP circles (Twig is included by default within Symfony), this plays in your favor compared to eZ’s proprietary coding ways of the past.

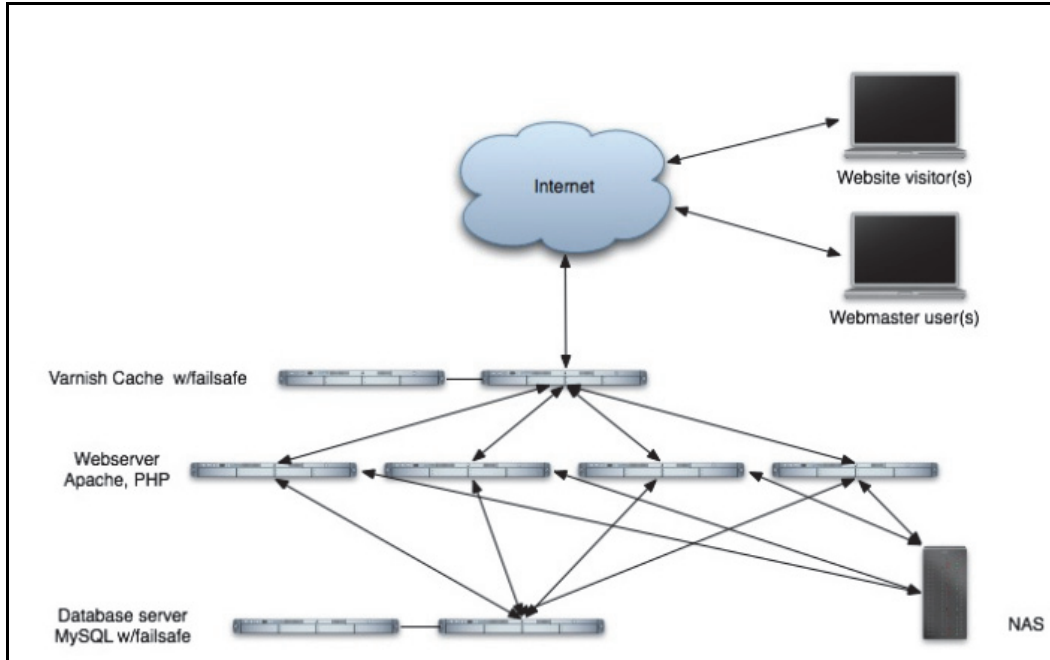


Figure 283. An example of a more complex, three-tiered eZ Publish Platform architecture seen in several larger projects. Be sure to calculate hardware and software costs carefully. Image source: eZ Systems.

eZ is a “coupled” system that “fries” dynamic pages from its repository, using its own display templates. Like with its more expensive competitor Sitecore, most eZ customers use the same server instance to both manage and deliver content. However, this option will not work very well if you’re using highly distributed, high-volume environments, where you will find the need to turn to more sophisticated methods of content delivery, such as clustering across multiple servers and using Content Delivery Network (CDN) services.

If you use LDAP, you can connect with eZ so that users can authenticate with their usual user name and password.

Like most open source tools, there is little to nothing in the way of native repository or system reporting services, even if you have access to raw logs. The tool will generate error messages, but they are too weak and cryptic to be particularly helpful.

On the authorization and authentication side, eZ’s security model is based on the following notions (Figure 284):

- **User** – User account represented as content object, which can be extended
- **Group** – Contains user accounts and other groups
- **Policy** – Collection of rules for accessing content that’s based on functions
- **Role** – Collection of policies assigned to users or user groups

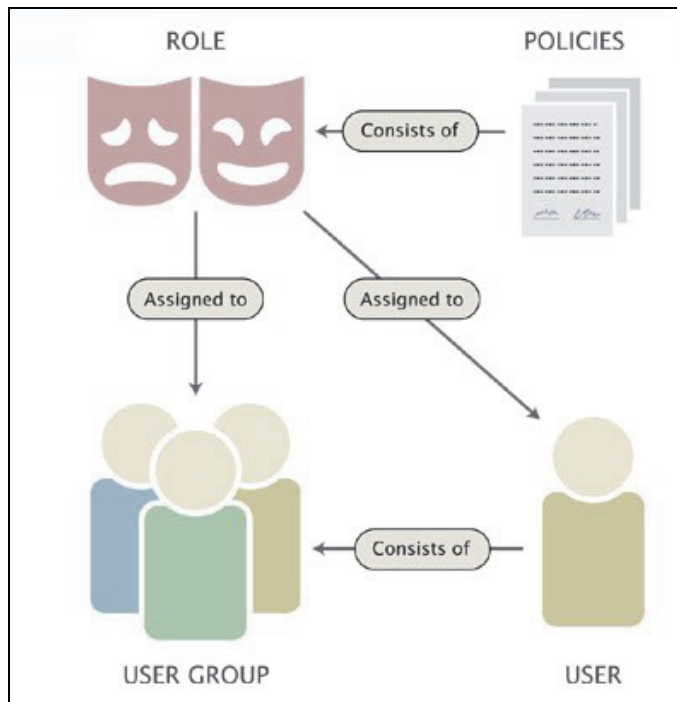


Figure 284. The flow of access controls and permissions in eZ Publish Platform CMS is simple on the surface, but is potentially complex under the covers. Image source: eZ Systems.

While this approach to access controls and security is frequent compared to other web CMSs, some eZ customers find it “cumbersome at first,” and conceptually hard to grasp due to its very fine-grained nature. Minor inconsistencies in this model, such as the ability to assign permissions on both the group and the individual user level can potentially create more mess than provide the perceived flexibility. eZ’s approach to complicating the basic content management functionalities is not surprising, given that eZ Publish Platform was designed by engineers. Therefore, keep in mind that it may not be easy to set up and work with this model for authentication and authorization.

Development

On the development side, eZ has been praised for extensibility and flexibility. “I have never been told I can’t do it in eZ and have to use something else,” remarks one eZ customer. eZ works with content objects rather than pages. Content types are called content “classes” in eZ, a sign of the system’s roots in a technical environment. Technical and non-technical users alike can define content classes using a browser-based interface (Figure 285).

The screenshot shows the 'Edit <Event> (0 objects)' interface. On the left is a sidebar menu with items like 'Collected information', 'Extensions', 'Ini settings', 'Languages', 'Menu management', 'PDF export', 'Packages', 'RAD', 'RSS', 'Search statistics', 'Sections', 'Sessions', 'States', 'System information', 'Upgrade check', 'Triggers', 'Link management', 'URL translator', 'URL wildcards', 'Workflows', 'Workflow processes', and 'Auth admin'. The main form area has a top bar with 'OK', 'Apply', 'Cancel', a 'Text line' dropdown, and an 'Add attribute' button. Below this is a header 'Edit <Event> (0 objects)' and a 'Last modified' timestamp. The form contains several sections: 'Name:' with a text input 'Event'; 'Identifier:' with a text input 'event'; 'Description:' with an empty text area; 'Object name pattern:' with a text input '<short_title|title>'; 'URL alias name pattern:' with an empty text area; 'Container:' with an unchecked checkbox; 'Default sorting of children:' with 'Path String' and 'Ascending' dropdowns; 'Default object availability:' with an unchecked checkbox; and 'Class attributes:' with a checkbox for '1. Full title [Text line] (id:316)'. Below this checkbox is a sub-form with 'Name:' and a text input 'Full title'.

Figure 285. To add a new attribute to an existing content type (class), there's a browser-based interface. The terminology is somewhat technical and it may take some time getting used to it. However, this interface is still more business-user friendly than working directly with definition files — the preferred method for some systems.

In short, the default content orientation is component-oriented. eZ Publish Platform enforces a separation between the content objects and its placement in the node hierarchy of the site. When a piece of content is put into a published state, it displays in one or more “locations:” one primary and any other(s) secondary. While this model is potentially powerful in terms of content reuse, it is daunting to most non-technical users, who are accustomed to a mental model based on folders. Adding yet more complexity is the concept of “Sections,” which is another way to organize content and can be used to restrict content access. It also can be used to determine which display templates should be used to display specific content.

Starting with version 5.0, eZ introduced a whole new set of options for developers, by adding support for the Symfony PHP application framework and (by association) the Twig templating engine. In order to access any of the new APIs, you'll to rely on these technologies. Be aware however, that even though there are more developers familiar with Symfony and Twig (as opposed to eZ's former proprietary tools), there aren't many of them who had applied this knowledge in the eZ Publish Platform CMS environment, beyond eZ's own R&D and Professional Services teams.

Traditionally, developers have had to write display templates using eZ's proprietary templating language, and its syntax is similar to the well-known PHP-templating library, Smarty. A presentation engine is available through “eZ Components,” which is eZ Systems' framework

for developing custom PHP web applications. Templates are written in a framework of cascading overrides, where a specific design overrides the default templates that come with the product. All these notions of eZ development still exist in implementations prior to version 5, and will eventually need to be migrated, despite eZ’s much touted backward-compatibility assertions.

Even seasoned PHP developers admit that the traditional notions of templating are not trivial in eZ Publish Platform, and you will require a complete understanding of the underlying system to feel comfortable here. This could mean what one customer described as an “almost vertical” ramp-up time of several months to a year or two, depending on your initial skill set and expertise. Among other potential limitations, you should know the restrictions on using pure PHP in templates. While there’s good reason behind this (separation of presentation from business logic), you cannot write templates in pure PHP; this only applies to eZ extensions, which then still have to be registered as a template block in order to deliver results of the logic contained in PHP functions. Even though the templating language is “darn flexible,” as one developer mentioned, you may feel discouraged by learning a whole new, albeit Smarty-like, language.

Concerning integration with third-party services, eZ Publish Platform hasn’t exhibited much success. Customers tell us that integration is one of the weakest points of this CMS, despite the presence of REST APIs that were introduced in version 4.5, which were rather shallow and limited in the early stages. You will want to test if the APIs meet your needs, and can extend the system and support external integrations and mobile solutions.

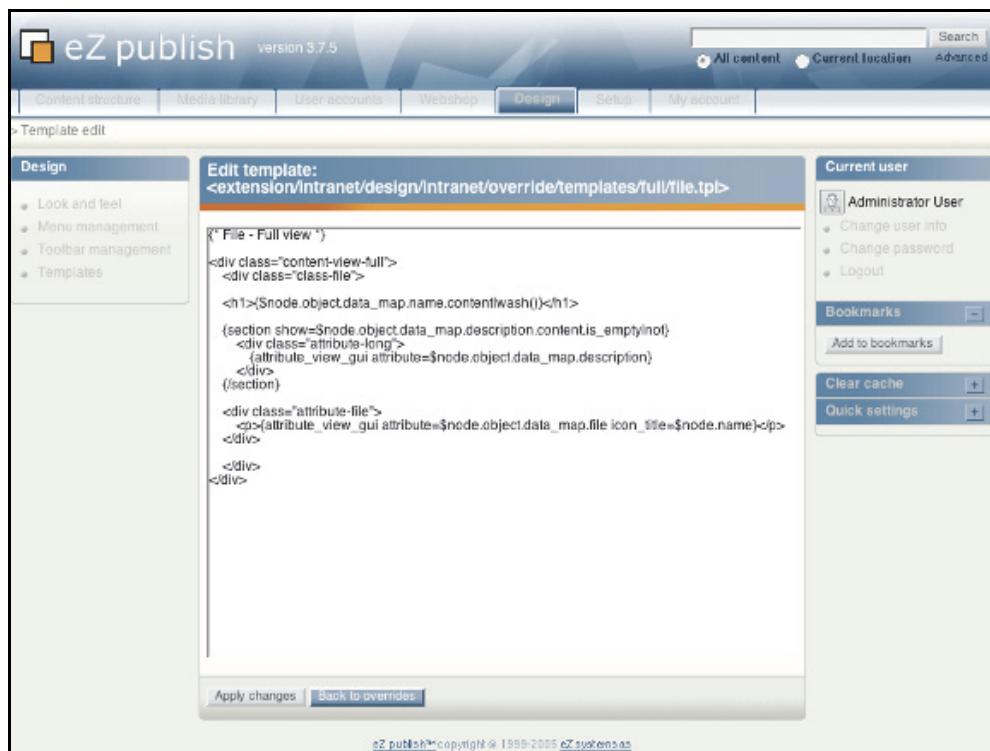


Figure 286. Templates can be edited directly using the browser, although most use PHP edit or Adobe as integrated IDEs.

Performance

While eZ Publish Platform used to be one of the slowest PHP-based web content management platforms, the PHP5-oriented re-engineering from v4.5 onward improved performance substantially. To be sure, the traditional eZ Publish architecture is still a heavyweight and processor intensive compared to your average PHP application. If you’re using the legacy templating, you should probably consider using a PHP accelerator, such as eAccelerator, that stores pre-compiled byte code, so that the runtime engine does not have to read the PHP code file from the file system with every page request. Also, by using PHP5’s native XML libraries and improved “include” system (Auto Loader), as well as some smaller optimizations, eZ Publish Platform customers say that they have seen a two-fold performance increase. To improve system response time, eZ Publish Platform supports four levels of caching:

- **View caching** – Caches the contents of the page
- **Block caching** – Cashes the page fragments
- **Static caching** – Caches the complete page in XHTML
- **HTTP caching** – Caches using third-party services like Akamai, Varnish, and Squid

As with some other systems, meticulous caching tweaks need to be configured and frequently tuned for optimal usage; otherwise, be prepared for serious hardware and performance issues. Of course, you need to learn what’s happening behind the scenes to use the advantages of this feature. There are many different ways to set up caching, and with many bells and whistles, administrators find it confusing in the early stages of working with the system. However, we’ve heard from some eZ customers that documentation has improved.

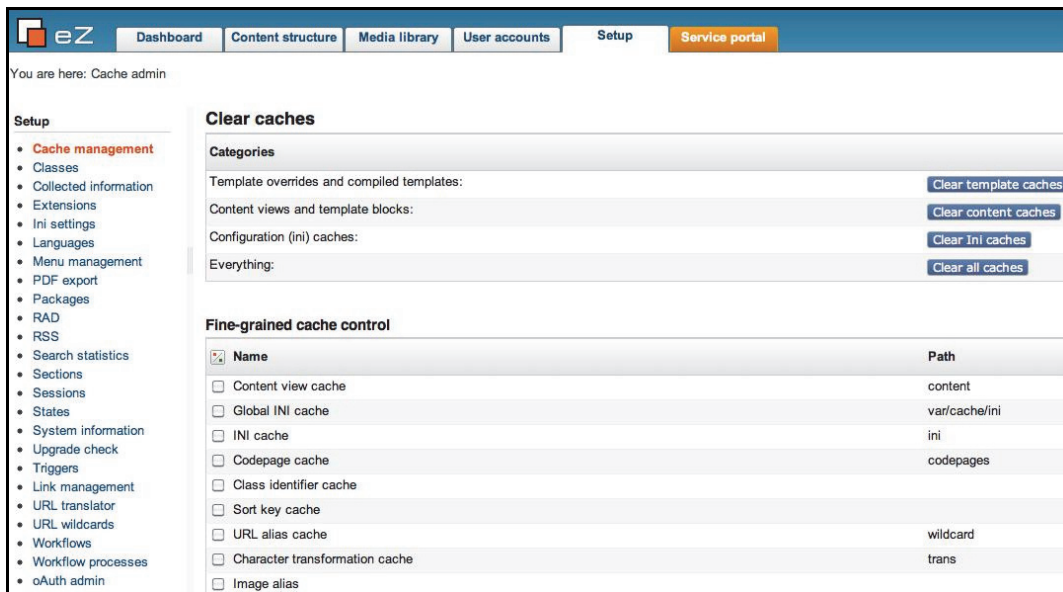


Figure 287. Cache management in eZ Publish Platform has an overwhelming number of options and fine-grained controls.

A word of warning: Editors can easily flush the entire cache using the editorial environment. This is quite nice for testing and for small sites, but on larger sites, this could considerably slow down your site if the entire cache needs to be rebuilt, one click at a time. This can be prevented by setting the appropriate permission control; be sure to do that.

Because eZ Publish Platform is still a resource-intensive application, and runs better on a physical server than a virtual machine, those looking for shared hosting providers may find this difficult with eZ. Note though that eZ has begun building its very own network of recommended regional hosting partners to try to address this issue.

Some customers report that they are using the ESI (Edge Site Include) standard (initiated by Akamai) to improve its performance in dynamic content assembly scenarios.

Scalability is another area to investigate — especially for scenarios with higher amounts of content. One customer reported hitting database limitations, and creating workarounds to manage live and archived databases with clustering, in order to circumvent the unclear way to archive without breaking the existing structure. “Shouldn’t be that we have to figure this out,” the customer concluded.

Content

Contributor Experience

There are several methods for adding content. You will use one of them, or a couple, depending on your role in the organization, as well as your skillset and personal preference:

- Front-end editing, working directly on the staging website with in-context editing. The front-end editing toolbar allows you to do certain tasks based on your permissions.
- Using the “Administration Interface” (the “back-end”) that has gone through a series of redesigns in prior versions, as well as the latest one, but is still very complex and rarely used by casual content contributors. The majority of non-trivial tasks will have to be performed here by power users.
- Using the Online Editor, a TinyMCE-based rich text editor for WYSIWYG editing. Keep in mind that some customers hit limitations with this interface, where WYSIWYG is only about 50% functional and gets you to “what you see is sort of what you get.”
- Using an integration with Open Office or Microsoft Office over WebDAV (“web folders,” if you are a Windows user)
- Using an integration with Adobe InDesign (implemented separately, not out of the box)

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

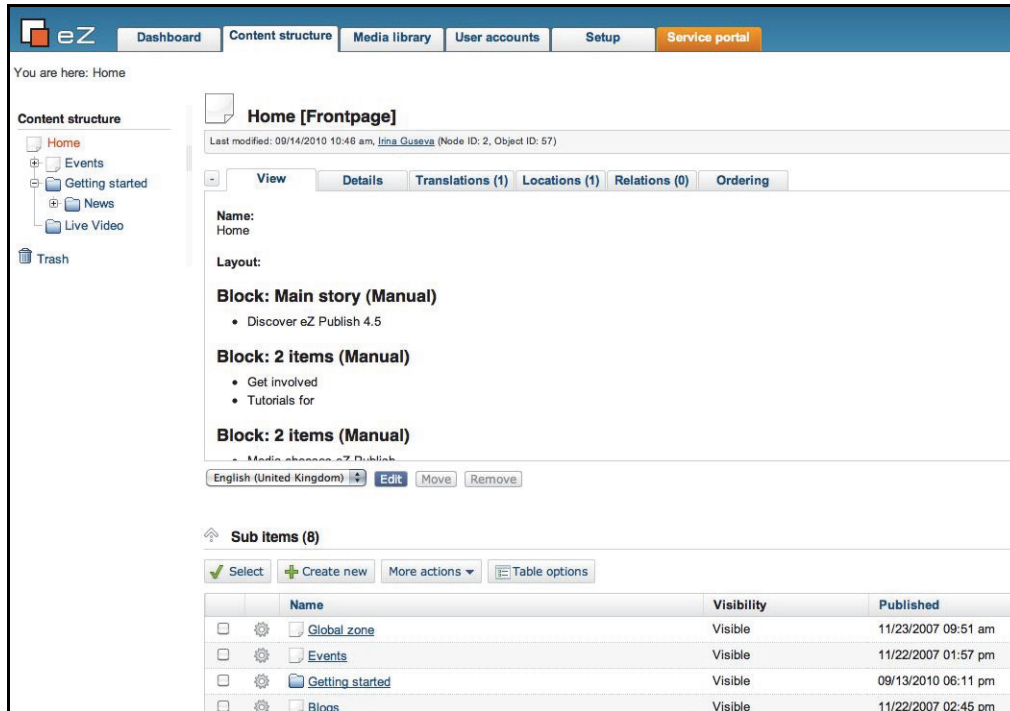


Figure 288. Content authoring and editing view under the Content Structure tab in the eZ Publish Platform admin interface. Some editors may prefer in-context editing, instead of using this interface.

The editorial environment is available in English but can be translated to other languages. There are 70 localizations available for download. These community translations have varying levels of completeness and version compatibility, however.

Contributing Content

Contributors who edit infrequently can work directly on the site using the in-context editing interface without going through the back-end. The front-end interface injects a small toolbar at the top of the page. The toolbar contains buttons to edit the page, add the page to another location on the site, or download the page in Open Document format to be edited locally using Open Office. The edit button reloads the page in the general public-facing page wrapper but with a web form, where the content is rendered.

Historically, eZ Publish Platform's user interface hasn't been highly regarded, but this started changing in version 4.4, which addressed some of the clumsiness and complexity. However, test it for yourself to make sure this is something your content contributors can work with on a regular basis. eZ's roadmap for the 5.x series suggests that the UI is due for a major overhaul. As with all roadmap promises, a large caveat emptor applies here.

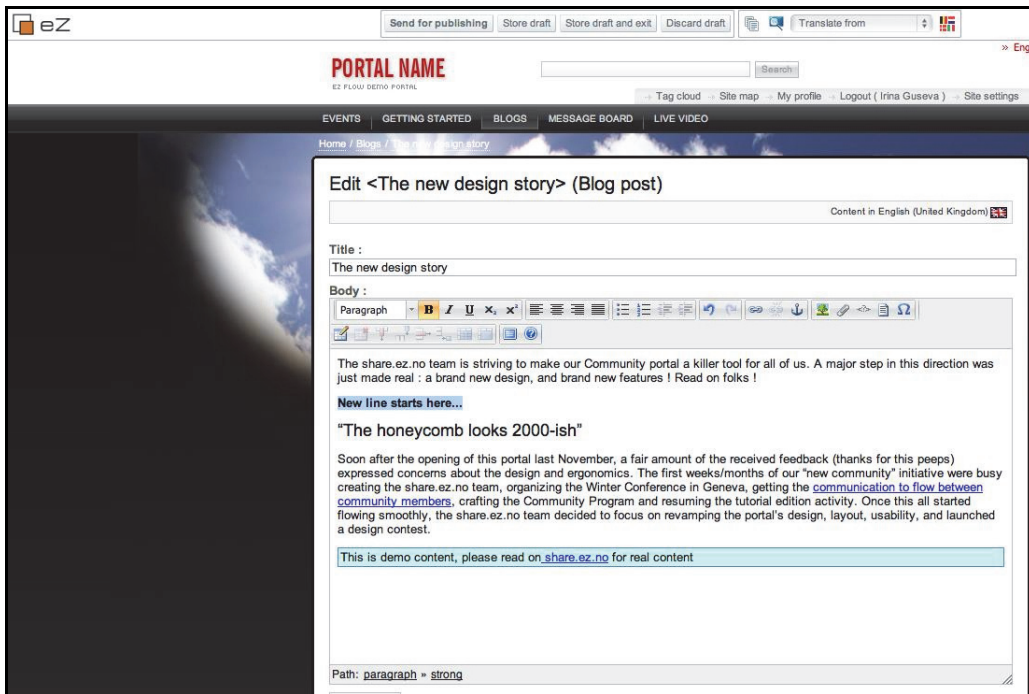


Figure 289. The eZ Publish Platform front-end interface provides an in-context editing experience to content contributors. A toolbar at the top of the page provides the option to send for publishing, or save as a draft — among other contextual options.

eZ Publish Platform uses the Open Office server-side libraries to handle Microsoft Office and Open Document file formats. In addition to extracting text to help with full text indexing, these libraries are also used in the content production process to transform content edited in Microsoft Word and Open Office Writer into structured content assets. From the Web Interface, the button to download the page in Open Office creates an Open Document (.odt) file out of the asset and starts the download. Different structured content attributes are put into sections of the document, so that they can be read back into structured attributes when the updated document is uploaded back into the system.

For rich text areas, eZ Publish Platform simplifies word processing formats, e.g., eZ Publish Platform filters out everything but bullets, tables, paragraphs, headings, bold and italics, and images. Image positioning is also simplified. While in Open Office, you can precisely size and position an image and how the text will wrap around it; eZ Publish Platform simplifies positioning to left, center, or right justification. Image size is translated to the closest predefined size alternative for the image. This helps to ensure that the layouts are more uniform and are W3C compliant (i.e., it doesn't retain all the garbage that a word processor can add). On the negative side, authors have the illusion that they have more control over formatting than they actually do.

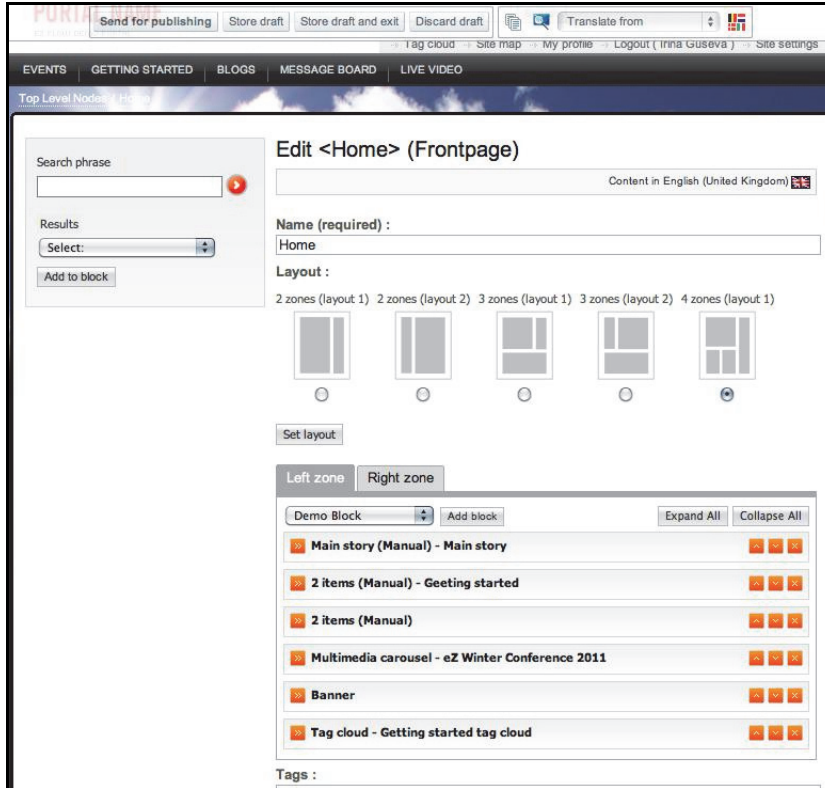


Figure 290. Editors can control page layout through a model of zones and blocks.

Scheduled content can be tested by a timeline feature that enables you to view how the page will look at a specific point in time — an essential adjunct for this sort of capability, and very useful for media properties, in particular.

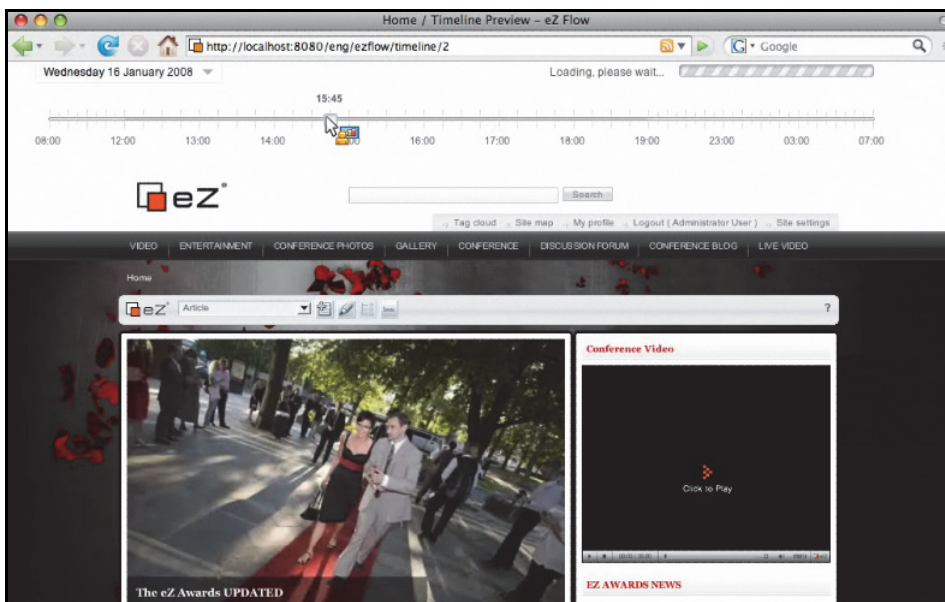


Figure 291. The “Timeline” feature shows how a page will look at a specific point in time.

eZ Systems has been adding video functionality, and several of their media customers such as the French Premiere and Elle sites use it heavily. eZ also introduced a live video feature that broadcasts video as it is being recorded and can be played back later. Video is broadcast in a low-bandwidth, Flash format similar to YouTube. There is also an iPhone friendly MPEG4 option. You can even use Apple’s Final Cut Server to publish video from Final Cut Studio to eZ (where Final Cut Server will handle asset management and workflow for video, with eZ providing the web publishing component).

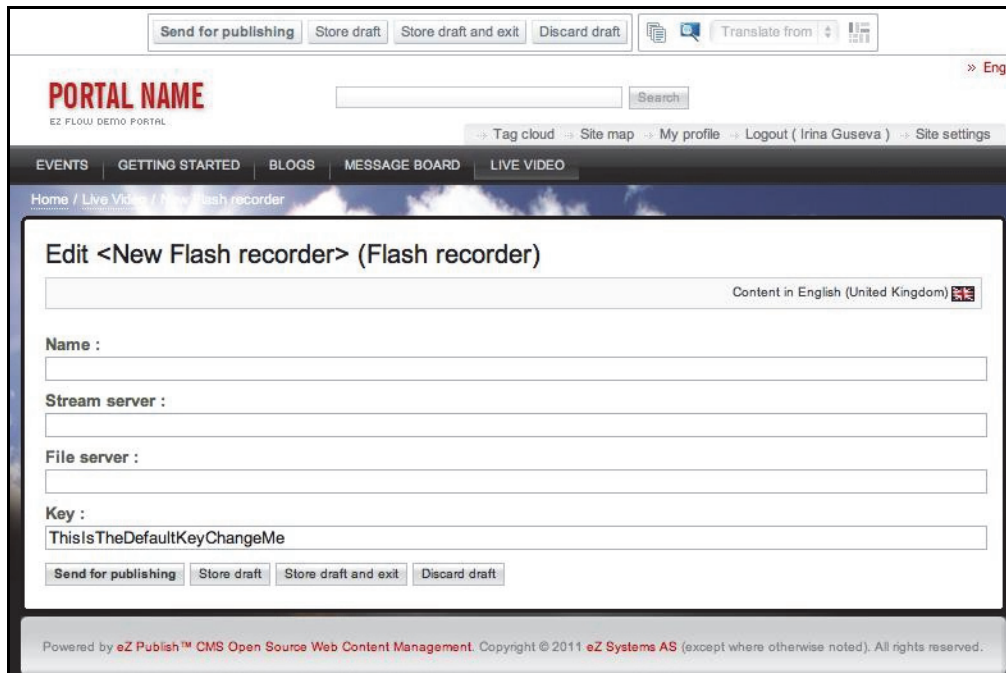


Figure 292. Setting up a Flash recorder class for live video still requires a lot of manual data entry.

For any semi-technical tasks: configurations, settings, localizations, etc., you will need to go to the back-end interface, which presents more power, but also more complexity. Despite eZ’s work to improve this interface, customers still report that it is not intuitive. In one situation, a customer completely locked it down for the majority of staff because “there was no way the editors could’ve figured out how to use it.”

Content Lifecycle

Workflow in eZ Publish Platform is primitive, and many customers describe it as a “weak point” in the product. While simple 2-step workflows may not be as painful, customers with advanced workflow requirements must resort to messy workarounds. One common approach is to overload the “sections” construct (that are normally used to organize the site and for access control and template selection logic) to represent workflow states. In other words, there may be a section for “editorial review” alongside a more traditional section like “news.” An additional problem is that a content object can only belong to one section at a time. Therefore, you will run into limitations such as not being able to send different types of articles to

different workflows, unless they are set up in separate branches. You'll have to get "a little creative," admits one of the implementers.

The "Object States" introduced in version 4.1 mean that the groundwork for more flexible workflows has been laid; objects can now have any state, such as "approved" or "translate."

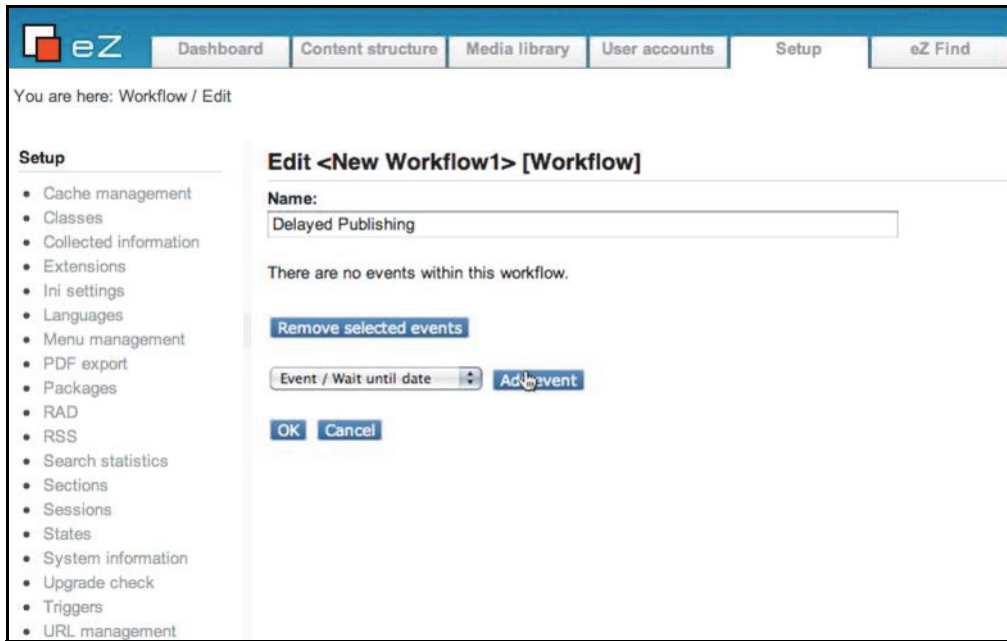


Figure 293. Workflow in eZ Publish Platform is primitive, and many customers describe it as a product "weak point."

Multiple sites can be created using the administrative interface and each site can have its own graphical design. In addition, different permissions can be set up for each site or even specific sections of a site.

The content model has good support for versioning and globalization. Here, eZ Publish Platform works best in a parallel model, where the content of different localized versions of a site is kept in sync. In eZ Publish Platform, each version of a content object can have multiple translations.

Users can be notified when a translated asset is updated. On the front-end, there are fallback rules to show alternate translations of an asset if the requested translation does not exist. This design offers the potential to manage a site with multiple translations centrally. Versions can be rolled back across languages, and if a translation does not exist, eZ Publish Platform can fall back to a close approximation. For example, US English when British English is not available. For non-parallel translation, it's recommended to use eZ Publish Platform's multisite hosting capability. However, this is still not an ideal product for globalization scenarios, given its dearth of native workflow.

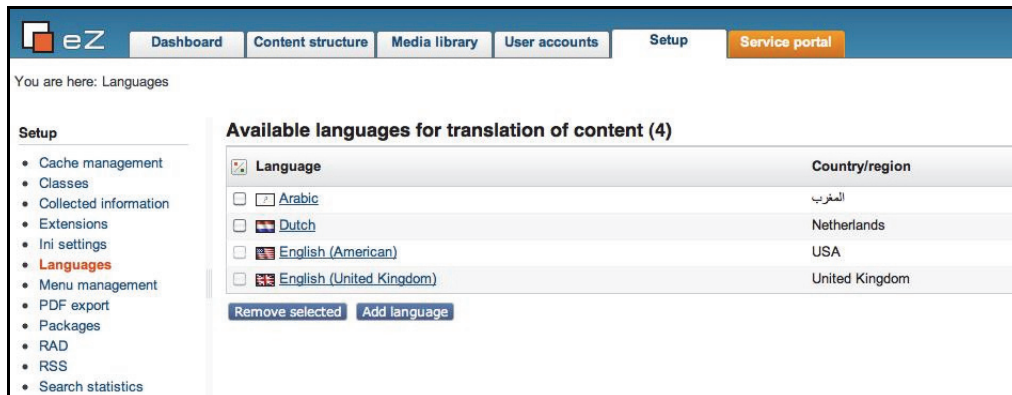


Figure 294. Language management in eZ Publish Platform enables you to add/remove the languages you want your site to support. However, this is just a configuration step, and all translations will need to be done separately, in a separate

Experience

Publishing

Just like Plone, eZ creates nice, short, user-friendly URLs by default, such as this one for their eZ Publish Now edition:

<http://ez.no/ezpublish/now>

The software can output clean code, which helps in both accessibility and SEO. In fact, eZ’s own website is one of the few among the vendors in this report that pass the W3C Validator for XHTML adhering to the W3C XHTML 1.0 transitional with flying colors. eZ Publish Platform also conforms to the accessibility standard “Section 508” of W3C WAI standard.

The asynchronous publishing feature was added in version 4.5 to accommodate larger enterprise implementations with a need to publish massive amounts of content concurrently by multiple editors. With this feature, you can restrict the number of concurrent publishing requests and queue up the rest of them.

eZ Systems generally falls short on the mobile and email delivery fronts. The product’s native approach to mobile is still somewhat limited with only two options of iOS SDK or the traditional mobile app development — even with the introduction of “eZ App Factory” module. The module allows users to populate pre-formatted generic templates with content, and then bake them into apparently native applications for iOS and Android. The module certainly puts a tick in the mobile box; however, experience with such generic OS templates has historically been very mixed, with a tendency to produce over-simplified and inflexible

Experience	
Publishing	
Standards Adherence	
Multichannel	
Mobile	
Digital Marketing	
Site & Campaign Analytics	
Testing & Optimization	
Segmentation & Personalization	
Social Media Integration	
Promotional Campaigns	
Community & UGC	
Workplace	
Collaboration & Networking	
Dashboard	
Ancillary	
Site Search	
Online Forms	
Module Ecosystem	

templating and variable device support — particularly with Android and its somewhat fragmented ecosystem. If mobile publishing is crucial in your scenarios, you’ll want to invest in a good deal of testing here.

Some customers hack config files to enable mobile detection; others chose to write custom handlers and template operators. Looking at a broader CMS market, eZ’s approach to mobile delivery is one of the most painful. While eZ has several customers running mobile apps, especially those in the media industry, you should carefully consider the effort it would take to complete a mobile implementation.

The Adobe InDesign integration, on the other hand, is implemented in a more sophisticated manner. eZ Systems is looking at print as the next big “money maker,” for customers who want to push content to the web and then pull it back for print.

Digital Marketing

Via acquisitions and partnerships, eZ has attempted to beef up its digital marketing offerings, adding Marketing Automation via a partnership with Net-Results. In addition, in 2011 they acquired recommendations-engine vendor YOOCHOOSE and analytics vendor odoscope.

The Net-Results partnership provides the “eZ Marketing Automation” rebranded integration into the eZ back office, so that campaigns — both for web multi-variance and email — can be created, managed, and monitored. This is a good start, but yet another almost-brand new integration that you’ll need to test thoroughly.

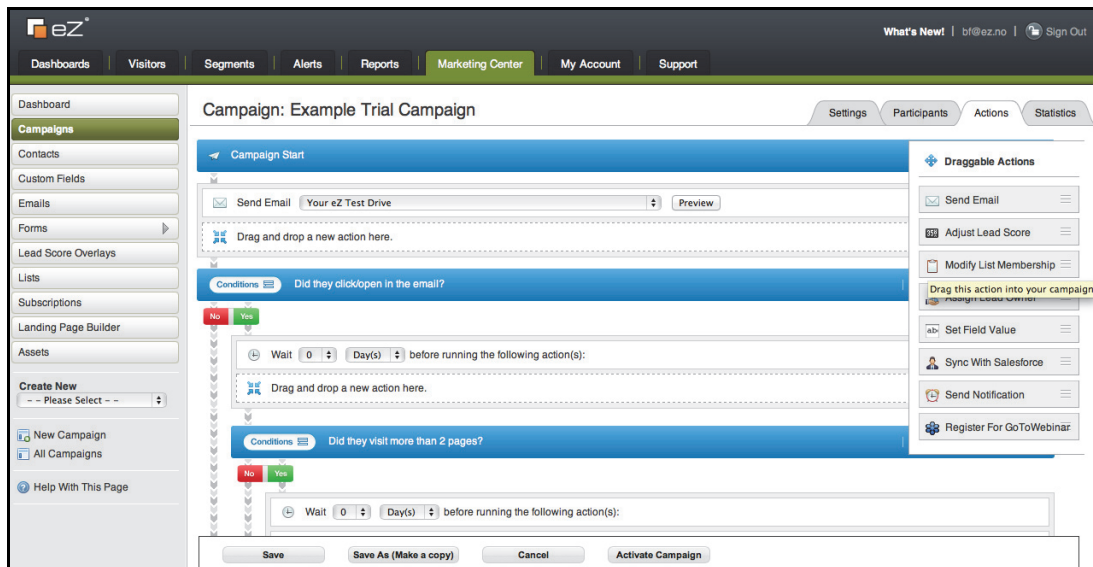


Figure 295. In a quest for CXM, eZ attempts built-in marketing automation in its CXM Platform, which is not always the best approach for the majority of use and business cases, where many organizations may prefer true integrations with best-of-breed marketing automations tools.

YOOCHOOSE now goes under “eZ Recommendation Services,” and it is designed to provide customized recommendations. The recommendations can be managed to produce a mix of forced placement and behavioral attributes that are dependent on specific configurations.

If you have blogging requirements, eZ Publish Platform has that ability, and you can host multiple blogs on the same site. The blogging functionality comes with the usual features for commenting, tagging, web feeds, and archiving. You can also create a tag cloud to visualize the most frequently used terms on your site.

On the extensions side, you will see wikis, polls, comments, forums and other expected social tools, similar to that of Drupal. One thing you should know here, though, is that eZ Systems leaves a lot of social media and networking integrations work in the hands of its community and third parties, providing little direction and showing little strategic thinking. One customer said that the vendor “dropped the ball” in this area of the product. There’s a separate e-commerce module, but it has been accepted with mixed feelings by customers and partners.

Ancillary Services

For search, eZ uses “eZ Find,” which uses Apache Solr (based on Lucene indexing). eZ Find makes good use of its tight bond with eZ Publish Platform, and can leverage much of the metadata of objects in the eZ repository. This means the add-on makes it relatively easy to build interfaces to display facets, refine searches, and let users drill-down into results. Many enterprise search products, being relatively unaware of the structure of a repository, will take a search result set and then find “clusters” of topics and display them. eZ Find has the luxury of being aware of what has been defined in the repository, and can use metadata to display facets across all objects.

Other vendors have started to develop similar integrations, but what sets eZ Find apart from most of its competition is a better understanding of enterprise search and the problems associated with it.

You should be very aware of the fact that the Solr component and its Lucene indexing engine are Java-based solutions, unlike the PHP environment used by the rest of eZ’s software. (In fact, eZ’s developers committed several additions to the Solr project, which facilitate the interfacing between eZ’s PHP code and Solr’s Java.) eZ maintains it is using the Java software for performance reasons. However, getting Java to run well will take a different skill set than operating a smooth eZ Publish Platform implementation — which is difficult enough in itself.

eZ hosts a well-regarded catalog of certified, third-party extensions available for purchase on its website. These extensions include Google Analytics integration, more sophisticated payment gateways, and user-facing applications like quizzes, advertisements, and newsletters.

Project Intangibles

The majority of eZ System’s engineers are based in Europe. Implementations typically get driven by a local partner. eZ Systems used to have a novel partner program based on a reward system, where partners would get points for being active and get rewarded with discounts.

Later on, this approach was heavily de-emphasized to the point of extinction. The partner program went to a paid membership — a classic model employed by many other vendors in the WCM space. The amount of money you will pay to become a partner depends on your

Intangibles	
Vendor Services	
Vendor Professional Services	<input type="radio"/>
Channel Partner Services	<input type="radio"/>
Support & Community	<input type="radio"/>
Strategy & Roadmap	<input type="radio"/>
Viability & Stability	<input type="radio"/>

training, competency requirements. The amount of revenue you generate can put you in a different partner level bracket.

Some customers and partners are not happy with eZ's community development. When there's a lack of community development and activity, it sends a worrisome signal to customers and partners that eZ Systems might be losing global momentum. Although the European community around eZ is widely held in high regard, other geographies have lagged.

Partners also complain that there's a need for better and more in-depth training. In its absence, they "have to endure six months of messing around" with the system to get a good grasp. This is a common complaint, and is a marker for you to vet partners very carefully.

The company's attempts to gain a foothold in North America have not been very successful: offices open and close; people come and go. As many vendors have already discovered, crossing the pond isn't as easy as it seems, particularly for a smallish company such as eZ Systems. The company recently relocated its North American office from Atlanta to New York in another attempt to break into the US market. Thus, despite years on the market, finding experienced partners can be a challenge, especially in North America.

To most customers, eZ Publish Platform will feel like a commercial solution with a centralized engineering team, customer meetings, and certified integration partners. Compared to Plone, where you can obtain support guarantees from a myriad of different systems integrators and freelancers, eZ offers this guarantee directly from the company by selling support from its headquarters in Norway, in the form of eZ Publish Platform Enterprise packages.

Historically, documentation has been spotty for eZ. Release notes are nowhere to be found beyond the most current version, so try not to fall too much behind in your upgrades. The user or technical manual is limited and not always up-to-date. If you're fluent in PHP, you can always take a look at the source code. For others, you can also turn to the community for help, as you would with any open source tool.

The development schedule is rather consistent with new releases coming out about every six months, with slight delays. While the product can be downloaded for free, there are two licensing options available:

- **GNU General Public License (GPL)**, where you can freely customize, modify, and distribute all parts of eZ Publish Platform (but must contribute improvements back into the project, if you choose to distribute all or parts of the product). This is free.
- **The eZ Business Use License (eZBUL)**, if you desire non-GPL licensing. While the actual software is based on what you get in the GPL version, each purchased copy of eZBUL includes bundled support for critical bugs and allows a different kind of setup to run websites.

Together with its different license models, eZ offers eZ Publish Platform Enterprise, with a choice of three different support and maintenance packages:

- **Silver** – 1-day response time, 2 support seats, for US\$15K/year
- **Gold** – 2-hour response time, 2 support seats, for US\$30K/year
- **Platinum** – 1-hour response time, 5 support seats, for US\$50K /year

Note that as a dual-license package, this software has been a creature of a single commercial parent company (eZ Systems) to a much greater degree than the "purer" open source platforms in this report, with the business model closer to that of Magnolia. However, the open source

version of eZ Publish Platform is identical to the commercially licensed version. Customers can get the commercial version for free with the purchase of eZ Publish Platform's support bundles. eZ Systems may charge support and maintenance fees for optional modules. An average deal size is at about US\$40,000; some larger deals go from US\$65,000 to 200K; a minimalist approach will cost you around US\$30,000. Some customers think eZ has overpriced its support options, since customers don't always need what the company offers, or sees little value in this proposition — especially from the perspective of budget constraints.

Fortunately, the company should be flush with cash right now. In the most recent round of funding, eZ Systems raised 3.3 million Euros from existing and new investors to support the growth strategy of a new CEO. This should be seen in a context of a history of turbulence at C-levels within the company that has (among other things) affected its relationships with partners. Some partners half-joke that every product release seems to be accompanied by a change at the C-level and a new CEO; it remains to be seen whether the new management will prove to be more strategic and more organized. Until then, we see concerns about company's viability on many levels.

Conclusion

Over its long tenure, eZ Publish Platform has proven itself a worthy contestant in the crowded open source CMS marketplace. What sets eZ apart from the others is its comparatively sizable number of corporate employees; the latest head count in 2012 was 105. Despite some departures in the light of turbulence in the C-suite, and continued general uncertainty as to the direction of the product, the company has a human resource capital that gives them an edge in developing service and support options.

Note however, that eZ has limited experience with larger projects beyond Western Europe. Experienced consultants exist, but they can be hard to find. A growing partner network and new expansion in North America may address this issue, but a prudent customer west of the Atlantic will monitor closely.

eZ has become increasingly commercialized over recent years, and the community is changing. Many systems integration partners seem to favor this transition (minus the increase in support fees), but more community-oriented developers may miss the "open source vibe" and grass-roots feel. For traditional firms and corporations evaluating eZ Publish Platform on their shortlists, the company still feels like a very recent college graduate, which is still trying to figure out what it wants to do "when it grows up." Despite recent overhauls in the C-suite, including the addition of a very ambitious CEO, eZ Systems as an organization still has a lot of work to do on the maturity front.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Hannon Hill: Cascade Server

www.hannonhill.com

Vendor at a Glance

Specsheet	Hannon Hill: Cascade Server 6.8 Summary
Geography	Primarily North America
What's New	<ul style="list-style-type: none"> • Enhancements to site publishing and previewing capabilities • Connectors to extend Cascade Server via third-party applications
Strengths	<ul style="list-style-type: none"> • Product offers a straightforward, no-frills, "back-to-basics" approach to web content management • System was designed with content reuse (at the level of XHTML chunks) in mind • Relatively clean Java-based architecture • Comparatively strong support for rendering content in multiple output formats for multiple target devices • Velocity templating technology (already familiar to some Java programmers) can be used in place of harder-to-master XSLT • Site Import functionality aids in migrating existing content • Online developer knowledge base is extensive and well organized • Company is small and accessible, priding itself on taking a "high-touch" approach with customers. The company has good experience in higher education and government sectors
Weaknesses	<ul style="list-style-type: none"> • Very limited support for scenarios involving interactive and/or personalized content • Programmer heavy, with limited customization capabilities for business users • Customization requires extensive knowledge of XML, XSLT, Velocity, and frequently Java • Templating subsystem feels unnecessarily complex • Lack of proper staging mechanism makes the product less suitable for larger enterprise customers • Lacks ready-made "social software" applications; social connectors only support one-way integration with Twitter & WordPress • Comparatively little market penetration or domain expertise outside of higher education, government, and health care • Relatively few experienced integration partners
Potential Fit	Informational Site
Unlikely Fit	Advanced Marketing Portal, Ultra-Large Single Site
Compare To	Ektron, Ingeniux, OpenCms, OmniUpdate
Operating Systems	Windows, Mac, Solaris, Linux
Repository	Databases: MySQL 5.0/5.1, MS SQL Server 2005/2008, Oracle 10g/11g

Specsheet	Hannon Hill: Cascade Server 6.8 Summary
Client	Browser: Internet Explorer 7+, Firefox 2+, Safari 3+, Chrome
App Platform	Java 5; Cascade Server is bundled with Apache Tomcat
Licensing	US \$60,000 for single server, unlimited URLs, or \$40,000 for single CPU
Ownership	Privately held

Summary

Hannon Hill’s Cascade Server is a basic XML-based CMS built on solid Java foundations. Founded in mid-2000, Hannon Hill has turned into a stable if not very sexy vendor. The same could be said for Cascade Server, which attempts to prove that relatively basic WCM functionality will meet most customers’ needs. Indeed, Hannon Hill has prospered in higher education, non-profit, and local-government sectors, where reliability and ease-of-use are key.

Cascade Server is well-suited for sites with small to moderate page counts (i.e., in the thousands to low tens of thousands), without heavily personalized or transactional content. The publishing model is page based rather than content component based; consequently, it’s difficult to (re)use the content at a highly granular level without a significant investment in XML/XSLT or Velocity. You can easily *repurpose* content in this system — i.e., output the same content to the web, as a PDF, or some other formats. Cascade server also introduced XHTML/Structured Data Block that enables reuse of content pieces (rich text as well as structured data) in multiple target documents without a significant amount of custom work. Consider Cascade Server for simpler scenarios where you need to perform basic, quick deployments of non-personalized, mostly static (or at least not super-frequently, updated) information. Also consider Cascade Server (as you would Refresh and other decoupled systems) when you have invested heavily in custom delivery logic and do not want a CMS that will interfere with the website tier.

Scenario Fits	
Simpler Site	
Informational	<input checked="" type="checkbox"/>
Microsites & Landing	<input checked="" type="checkbox"/>
Mid-Range	
Basic Digital Marketing	<input type="checkbox"/>
Mobile Site	<input checked="" type="checkbox"/>
Community Oriented	<input checked="" type="checkbox"/>
Complex Site	
Advanced Marketing Portal	<input type="checkbox"/>
Global Enterprise	<input type="checkbox"/>
Multichannel Publishing	<input checked="" type="checkbox"/>
Ultra-Large Single	<input type="checkbox"/>
Digital Workplace	
SMB/Departmental	<input type="checkbox"/>
Global/Enterprise	<input type="checkbox"/>

Introduction

Hannon Hill is one of the industry’s younger companies, founded in mid-2000 and incorporated in January 2001. Headquartered in Atlanta, Georgia, USA, the firm has 25 employees and a smattering of regional channel partners. Company growth is based purely on cash flow (there is no venture-fund backing), and — unlike former competitor Serena Collage

— has traditionally kept a clear focus on a single product, its Cascade Server CMS. In November 2008, the company decided to test the SaaS waters with a hosted service — called eCrowds — featuring interactive blogs, wikis, forums, and “idea exchanges.” The product never took off and the company withdrew it in 2009.

Hannon Hill likes to boast that its system is “as easy as web-based email.” It feels about as simplistic, too, with not many rich or complex features out of the box. That’s not necessarily a bad thing, but it does speak to why Hannon Hill’s customer base is weighted toward higher education and public-sector customers: sectors where the ability to assemble relatively simple content quickly and easily (without administrator, developer, or Help Desk intervention) is paramount. The product supports simple, straightforward content scenarios, where easily assembled pages are pushed out “fully baked” to the delivery web server on a relatively infrequent basis.

Version 6.8 was released in early -2011, adding features aimed at improving publishing and previewing capabilities. However, the company’s approach has been to focus on the basics. That remains unchanged.

Technology

Technical Administration and Security

A Java-based system, Cascade Server was originally constructed around the Apache Cocoon Web publishing engine, but after some difficult early years, Hannon Hill judged Cocoon’s XSLT pipeline to be too slow and discarded it in favor of a Struts-based architecture.

Cascade Server now runs on the Spring framework, with the well-known open source Hibernate module as the persistence layer. The payoffs have come in better performance, better load-balancing, and a host of other benefits. Hannon Hill says the server will run on any J2EE application server. However, it’s only been tested with Apache Tomcat, which is bundled with it.

Cascade Server is completely XML based, meaning content is stored and managed as XML (or XHTML) “chunks” within the chosen relational database (most commonly MySQL). Administrators use a data definition tool to define content fields in the content manager’s interface, which in turn defines how the content is stored in XML. Developers then use Velocity templates (or XSLT, or both) to specify how content is rendered.

Despite the fact that content is stored as XML fragments, Cascade Server is fundamentally a page-based system. Pages are transformed to XHTML via XSLT or Velocity, or to PDF via XSL-FO, enabling the rendering of different views of the content objects. (In fact, Hannon Hill’s own site allows you to view pages in HTML, PDF, WML, and other formats.) If you don’t have XSL or Velocity expertise in-house, note that the learning curve can be steep,

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

especially for XSL. Hannon Hill customers confirm that the time it takes to realize the benefits of this CMS are highly dependent on staff knowledge of XML technologies.

Cascade Server stores everything in the database including documents and images. While page content, block content, and metadata is stored as text in the database, files are stored as Binary Large Object (BLOB) data. This not only could make your database bulky if your site has heavy usage of rich content (like images and videos), but it also means that text file content are not searchable. To work around this, Cascade Server puts the text file content in the Lucene index so it is searchable along with the other text content in the database.

Fundamentally, this is a decoupled system. Cascade Server grabs XML content and creates “baked” pages (using XSL transformation or Velocity scripting) for delivery to the web server. Whole pages (not components) are managed and previewed in the CMS, though those pages do break down into “blocks” that can be applied to multiple pages.

This sort of page-based “push” model is clean and simple and thus is a good fit for a non-Informational Site or a Departmental Intranet, including the types of scenarios found in university settings. Hannon Hill says that customers in certain industries such as health care and government actually prefer a bake-and-push model for compliance reasons.

Content can be pushed out (manually or via scheduled jobs) via FTP, WebDAV, or other typical methods — all of which are non-transactional. Cascade Server provides you with a few options to manage what content gets pushed out. You can:

- Publish the entire site at once
- Publish only content that has changed since the last publish job
- Publish a predefined set of content using Publishing Sets
- Publish specific content via workflow triggers

Alternatively, you can pull content out of the system programmatically via SOAP-based Web Services, and push content to a database. Hannon Hill provides the schema and a set of PHP and .NET libraries.

Out of the box, Cascade Server doesn’t provide support for multiple environments like development, testing, staging, and production. As a workaround, you can create multiple destinations to simulate these environments, and do a selective publish. The company claims that the workflow-based push can also publish to a particular destination. However, a customer has mentioned that the workflow will actually publish content to all destinations. This is something to test if this is one of your critical requirements.

As another workaround, you can employ “Live View.” This enables you to view the published page from within your dashboard for editorial validation. While this displays the page with server-side scripting enabled, that content is already published, so this is a post-facto check and not a proper staging preview.

Publishing to multiple sites is a bit funky, but it’s better than it was before. Parallel Publishing (a feature introduced in 6.8) enables each site in the system to have its own independent publishing queue; however each queue can still only publish two assets at a time. Although this can be a limitation if you operate large sites, it is a significant improvement over the previous versions where the entire system had only one queue for all sites.

Cascade Server supports groups (named collections of individuals) as well as roles (permission-sets mapped to groups and individuals) through a fairly elaborate — but visually

easy — user interface. Interestingly, roles can now be scoped by site, where a given user’s privileges differ across sites, such that an Author in one site can serve as an Editor. The system’s rights-management capabilities are more than powerful enough for the vast majority of customer types that Hannon Hill sees. More important than sheer feature-richness here is the fact that the rights-management functionality is specifically designed to accommodate the multisite, delegated-administration use cases that Cascade Server tries to serve.

For authentication, the system supports Active Directory as well as ordinary LDAP servers, and if LDAP isn’t what you want, you can either use Cascade Server’s default authentication or choose to develop a custom authentication mechanism by using the tool’s authentication API. However, if you are a large enterprise with multiple LDAP servers, you’ll be constrained by Cascade Server’s limitation to support only a single LDAP server.

System reporting is not a strong suit but not poor, either. System activity audit trails can be viewed in the administrative UI (in the browser), filtered by user, group, or asset.

Development

Hannon Hill exposes a Web Services API for most CRUD (Create, Read, Update and Delete) operations available in Cascade Server. While customers seem to be enthusiastic, not many seem to have actually exploited the API. Customers using these Web Services are quite happy with it and have developed simple tools to facilitate administrative tasks, as well. Scripts are provided for importing both structured and unstructured content, but in reality, content cleansing and reorganization are often necessary after employing this functionality.

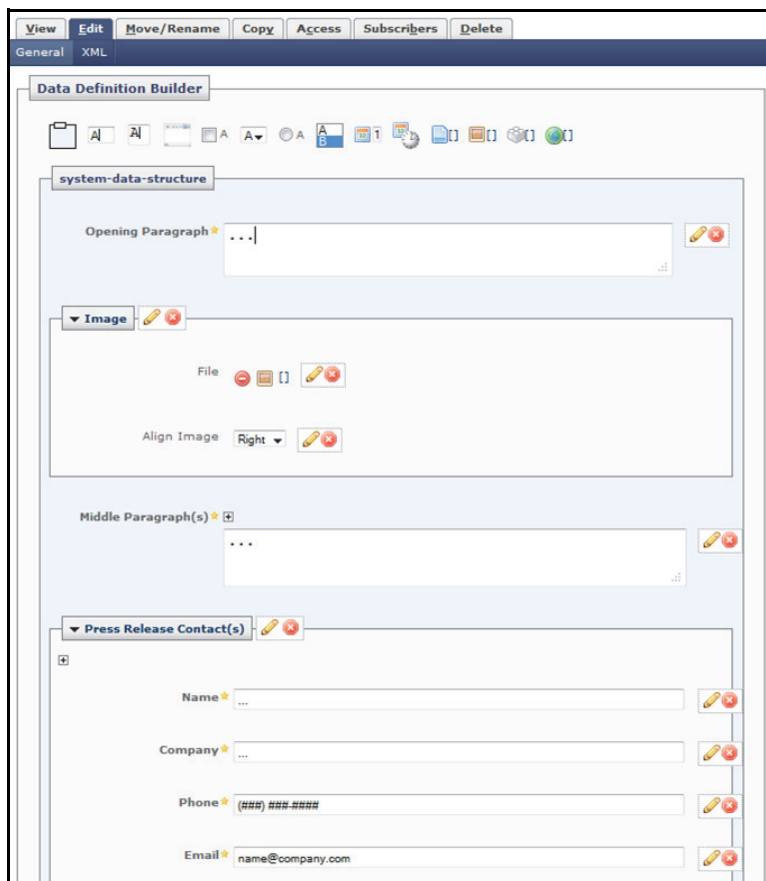


Figure 299. Entry templates are created via a data definition interface that doesn’t require coding.

Hannon Hill calls all managed content items “Assets.” These can represent things like pages, links, and folders. Content entry templates with standard metadata fields based on Dublin Core are provided out of the box, and custom entry templates can be created easily via a data-definition interface. Presentation templates can be created/imported from an existing static site, or you can upload or paste an HTML template into the template editor and save it as XHTML. Hannon Hill provides a set of custom XML tags to populate the templates and subsequently, the pages with the appropriate content. Once you have a set of templates, you create what are called “configuration sets,” which tie the content to the output type and group several templates together to form a single page.

You can then create formal content types as first-class objects, which combine metadata configurations, data definitions, and output types into one class. In any case, you’ll almost surely need an experienced developer to create and modify templates.

Presentation templates are created, viewed, and edited in an interface similar to the one content managers use to view XML content.

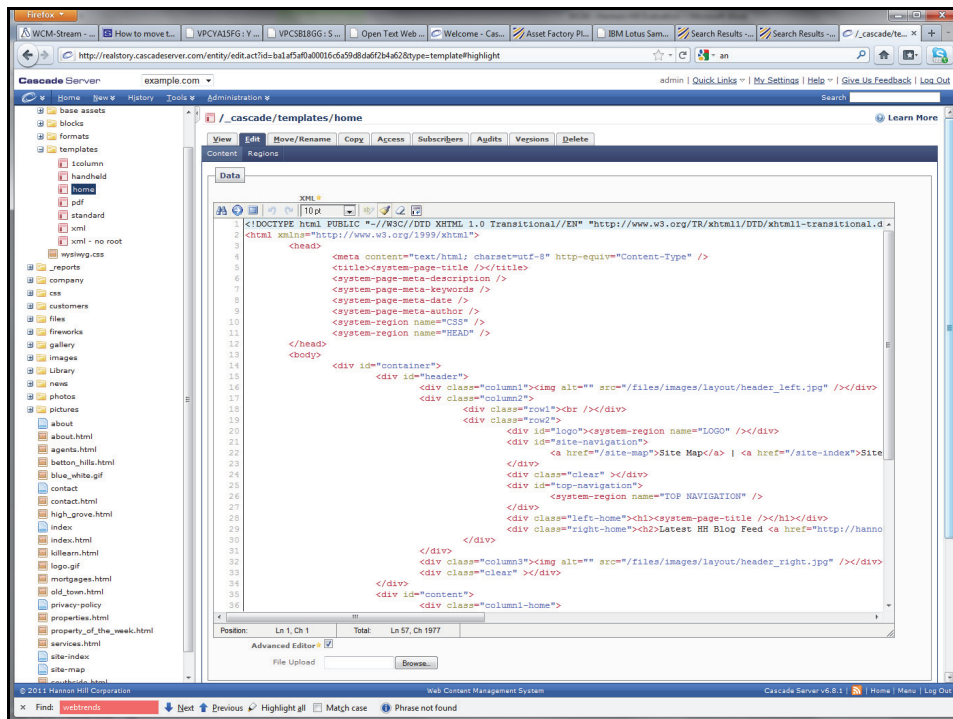


Figure 300. Cascade Server’s template editor.

Traditionally, one of Hannon Hill’s main value propositions has been “up and running in under a month.” To make this possible, you can either import a zipped file of your existing website or use a third-party tool, HTTrack, to get a static, local copy of the site. With this facility, an existing site containing thousands or tens of thousands of pages can be imported into Cascade Server relatively quickly, and immediately provides version control and other basic repository services. In the overall scheme of things, little other value is realized by this initial step, because the information value of the content remains locked up until it’s converted to an XML type the system can use.

Performance

Depending on how many pages you update, how often, and how complex those pages are, a full bake-out can be resource intensive. It depends on many variables, though, so plan to do some investigating. There is no pat answer to performance-related questions. On the plus side, Cascade Server does support clustering and load balancing, but these are somewhat complex to implement, and a poor implementation can actually impede performance. Again, plan on doing some homework here, and budget for adequate technical help.

Content

Contributor Experience

Cascade Server’s user interface is among the most Windows (XP) -like of any CMS, not only due to the folder browsing and dashboard paradigms, but also thanks to the “Windows blue” color palette and iconography (Figure 301). Access is completely browser-based and requires no downloads of applets or ActiveX controls. Cascade Server uses the JavaScript-based TinyMCE rich text editor, which is not known for its great Word conversion support but can enforce XHTML compliance. Authors do not have access to “favorites” but your default dashboard will show all check-outs and recent documents.

If there is a downside to Hannon-Hill’s simplicity-is-king approach, the UI lacks some of the in-page interactivity that you’ll see in many competitors in this class. The company has been sparing in its application of AJAX, largely for accessibility reasons — very important for its education and government customers — and as a result, you may find yourself popping more windows and/or clicking more. At the same time, Cascade Server can boast a single, web-based interface for everyone: administrators, developers, and content contributors — something few vendors can do.

Cascade Server is internationalized and out of box, it supports eight international languages; note, however, that Hannon Hill remains focused primarily on North America.

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

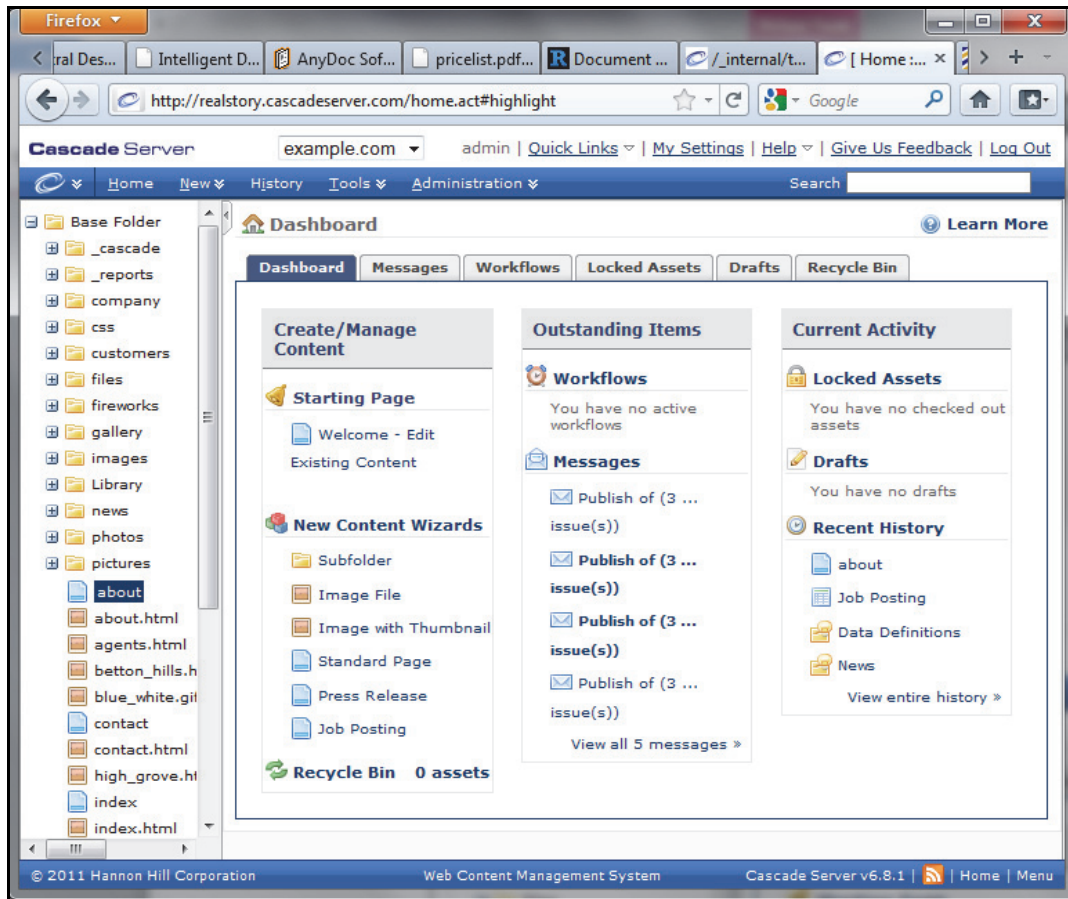


Figure 301. Cascade Server’s look-and-feel is generally uncluttered and Windows-like. To navigate to the Image Gallery, it often requires finding commands in deeply nested

Contributing Content

In-context editing is straightforward. Editable items on a rendered page are indicated by means of tiny icons (Figure 302). The logged-in contributor will see the icons only on items for which he or she has editing rights. If you click into one of the icons, a WYSIWYG editor displays. You can add or edit content and perform a Save (or even “Save As Draft”) at any time to preview your work.



Figure 302. The permissions-based, in-page editing UI is easy to learn.

This has not traditionally been a great system for producing and managing highly granular chunks of content destined for reuse in highly compositional pages. If you're in the business of presenting dynamically assembled inventory lists and catalogs, or bids, contracts, highly compositional technical documents, etc., be advised that while you can *make* the system do these things, it's not the product's sweet spot.

In theory, you can create content independently of any given website and other pages can "subscribe" to it, but the system does not provide the same convenient services for tracking and previewing as you find in many other XML-based systems.

On the plus side, within Cascade Server, you can toggle between different content outputs. Specifically, you can preview any single content item in all its different potential renditions (Figure 303); this is something few other systems enable so easily. Also, the product has some basic services for supporting multiple sites off the same instance. However, these mostly revolve around the variable permissioning and roles discussed above, and the attendant ability to delegate site management. This clearly comes in handy among Hannon Hill's higher education clientele (one university customer runs 96 websites off Cascade Server).



Figure 303. Once content is saved, it can be viewed and tweaked in various formats; see the “Outputs” bar on the top-left side.

The image gallery functionality enables content managers to drag-and-drop multiple images into the interface and assign metadata to them, as well as zip old images into an archive. However, accessing images is less straightforward and requires that you pop several windows. Cascade Server replaced the old image editing utility with a neat little custom utility built using a combination of Sanselan (from the Apache Commons project) and Java Advanced Imaging API (JAI).

Repository search is workmanlike. If you are trying to search non-text, non-HTML content in the repository for editing, you may not be totally satisfied with the plain-vanilla instance of Lucene that ships with the product; there are no filters for things like Word documents.

Content Lifecycle

Workflow is relatively simple, with an almost wizard-like UI (Figure 304); alternatively if you’re familiar with XML, you can create custom workflows from scratch. While it’s possible to create conditional workflows, there’s no functionality to support parallel flows with joins, complex retry scenarios, or anything more than straightforward serial approval flows. This is not unusual in this class of CMS products, of course, and customers seem to agree with Hannon Hill that when it comes to workflow, “good enough is good enough.” One customer told us that he’s actually glad that the workflow system *isn’t* more powerful. “It’s easy enough to get in trouble with it as it is,” he noted.

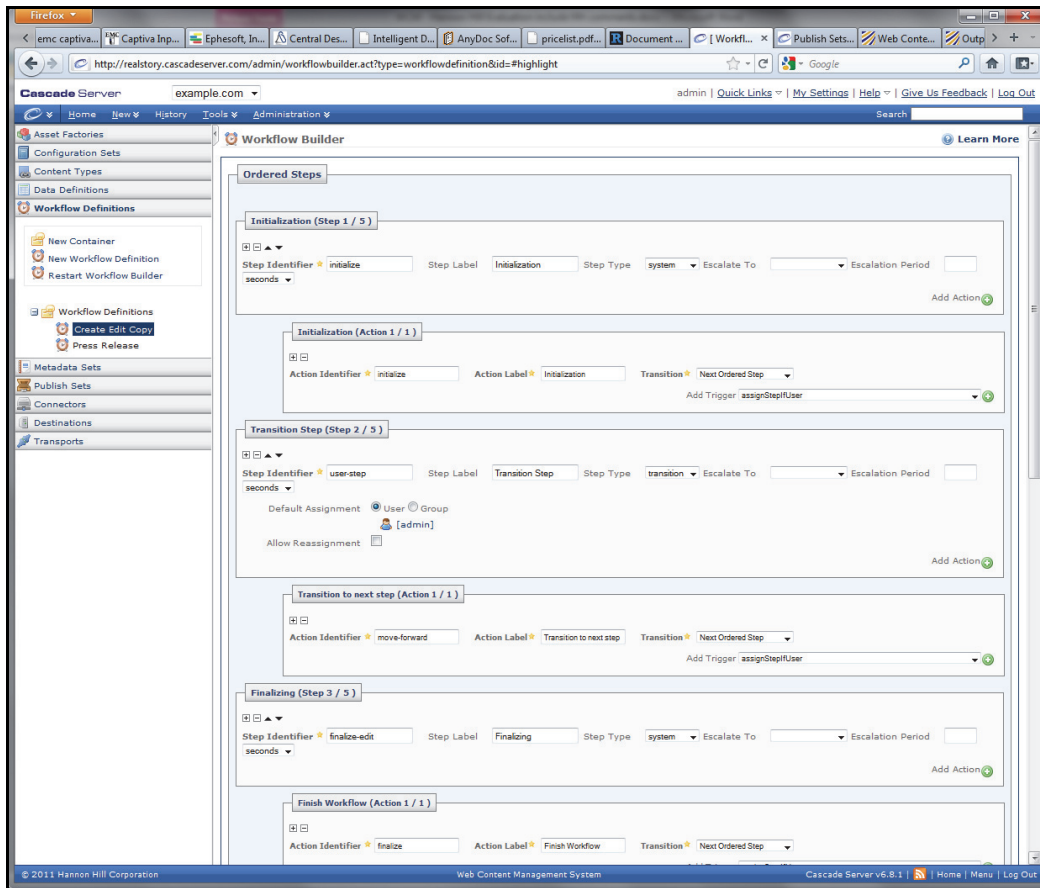


Figure 304. The Workflow Builder interface is fairly easy to understand.

Nevertheless, the lack of a visual (canvas-based) flow editor and the absence of advanced capabilities (like fan-out/fan-in), could prove to be problematic for some use cases, such as multinational publishing or companies that require heavy compliance or legal review cycles. Again, if you have anything more than simple workflow needs, examine this area of the product in depth before researching other areas.

From a workflow reporting standpoint, Cascade Server provides basic information around usage (user audit trails) and content auditing. This is useful, but while the information can be exported for use elsewhere, the native reporting functionality isn't as rich as that of comparable hosted solutions in this tier, such as CrownPeak or Upland Software.

Cascade Server has native Unicode support and canned workflows to facilitate the translation process, but lacks extensive experience in the domain of multilingual sites. If you're going to localize content in foreign languages, don't be too concerned with Cascade Server's other features until you verify that it can meet your needs in this potentially tricky area.

Cascade Server supports versioning, allowing editors to view and track changes to these assets. However, at least one customer complained that there's no option to disable this, which can dramatically increase your storage needs, especially on media-heavy sites. Additionally, if you need to compare two versions, you'll have to do it manually; there's no visual compare.

In terms of retention services, Cascade Server lets you set expiration dates for time-sensitive content. It will then automatically unpublish content at the expiry date and automatically move it to a folder that you have previously designated.

Experience

Publishing

The system produces search-engine-friendly URLs by default and offers automatic link management (so that links are updated when an asset moves or is renamed).

Digital Marketing

Cascade server offers a visual UI to configure interactions with web analytics to see how your end product is performing. This is helpful, because you can pull analytics data from an application like Webtrends or Google Analytics, display it inline within your CMS, and make any appropriate changes within the same UI. Very handy, although the product doesn't provide more specific, item-by-item overlays the way SDL and some other products do. Aside from that, this system devotes very little attention to features in the realm of experience management and pursued by many WCXM vendors.

Hannon Hill has not designed Cascade Server to provide personalized, digital marketing oriented or interactive content capabilities. More generally, it's not strong at dynamic content rendering.

You can ingest external RSS feeds, and then apply XSLT to style this content as desired.

Cascade Server's social media integration capabilities are fairly light. The extent of social media integration is mostly limited to using asset types called Connectors that enable integration with third-party products like Twitter and WordPress. This allows you to publish page content to your Twitter and WordPress accounts. This however is a push-only integration, which means that you can only publish content to these social media sites using these connectors, but you cannot fetch content. As a workaround, customers have used feed ingests to pull content from these sites to simulate bidirectional interaction with social media sites, but this approach is far from ideal.

Ancillary Services

As with internal (repository) search, Cascade Server defaults to a rather limited build of Lucene for site search. Cascade Server is not known for its rich assortment of micro-applications. If you want interactivity, you have to code it yourself and embed in the pages you push to the delivery tier. (In the process, you need to make sure that content authors are not messing with code, and they will not be able to preview a virtualized page fully until it is pushed to a staging server.)

Experience	
Publishing	
Standards Adherence	
Multichannel	
Mobile	
Digital Marketing	
Site & Campaign Analytics	
Testing & Optimization	
Segmentation & Personalization	
Social Media Integration	
Promotional Campaigns	
Community & UCG	
Workplace	
Collaboration & Networking	
Dashboard	
Ancillary	
Site Search	
Online Forms	
Module Ecosystem	

Vendor Intangibles

Hannon Hill typically has two or three “dot” releases a year where they focus on introducing new features, along with a monthly support release to fix bugs. Nevertheless, the releases do not frequently represent major changes; as such, they are rarely traumatic for customers. Hannon Hill has a quasi-democratic voting system where customers recommend new features for future releases. While those customers have been happy about the product’s stability, overall Hannon Hill seems a bit behind the curve in adding new functionality to the tool.

The knowledgebase and message board for developers make for a decent (if small) community around Cascade Server. There’s also a university webmasters’ mailing list where Cascade Server is often discussed at length. In addition, the company has invested steadily in making its online knowledgebase more thorough and easier to use.

Hannon Hill has also been attempting to become more responsive to its customers’ various technical needs by beefing up its training (which had been reportedly lackluster), improving its online knowledgebase and by hosting customer conferences in Atlanta.

Some conference sessions are recorded, and the company kicked off “CMSTube.com” to host the various videos, covering subjects ranging from “Advanced XSL” to Roles and Workflows. Customers report very responsive and reliable support, thus far.

Note that in all likelihood, you won’t find an experienced local Cascade Server integrator. The company can point to a couple of active integration partners in the academic space, but most other partners seem to have emerged from one-off projects, and you can’t call them experts. Hannon Hill does not have a partner certification program, and its training sessions — held in Atlanta — are relatively infrequent.

Cascade Server pricing is \$60,000 for a single server with unlimited CPUs including support. Maintenance, support and product upgrades are bundled together and cost 12 percent of the annual license price. Curiously, Hannon Hill’s most common license is a single-CPU server license for \$40,000 with unlimited sites, content, and users. It’s unusual to see CPU pricing at this tier, and for some customers, this approach could get pricey.

Hannon Hill does not have a SaaS offering; however — like Ektron and EPiServer — they do offer a hosted solution that starts at US \$30,000 per server. Government and non-profit customers can obtain special discounts.

That said, customers report that support seems responsive and — as with most small software companies — senior leadership is accessible. Email support is usually answered within the hour. At twenty-five employees, Hannon Hill is not the smallest vendor in this report, but is certainly not a large one, either.

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

Conclusion

Hannon Hill remains “steady as she goes.” The company’s founders have been focused on other, higher-growth opportunities, but Cascade Server retains a reasonable customer base, especially in higher education, where its no-nonsense appeal has gained significant traction.

To be sure, Hannon Hill is pricier than other tools with similar functionality (such as Ingeniux and OmniUpdate). Still, customers seem happy with the product. It apparently fills the “80” part of the 80:20 equation in a better-than-okay fashion for a certain segment of the market where, in fact, “good enough is good enough.”

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Ingeniux: Content Management System

www.ingeniux.com

Vendor at a Glance

Specsheet	Ingeniux: Content Management System 7.5 Summary
Geography	Primarily North America
What's New	<ul style="list-style-type: none"> • Support for in-context editing • Ability to create workflows visually • Single cross-browser client for all tasks
Strengths	<ul style="list-style-type: none"> • Emphasis on structured content and XML makes Ingeniux well suited to repurposing content to multiple formats or channels • Supports a variety of configurations (coupled, decoupled, n-tier) and publishing styles (fried, baked, or mixed) • Only one contributor UI for all platforms (Safari, IE, and Firefox) • Supports cascaded authentication enabling simultaneous authentication for identity providers • Simple, page-oriented approach appeals to non-technical end users • Search-engine friendly URLs • Company has a reputation for providing good support
Weaknesses	<ul style="list-style-type: none"> • Search performance can be poor on large sites • The collaboration module (Cartella) has features not present in the CMS • Templating still requires coding and lacks a visual or drag-and-drop mechanism to create layouts • Working with content types requires developers • Pace of CMS product development has traditionally been sluggish • Company is very small, with limited community, and has no presence outside North America • Emphasis on XML and structured content is overkill for simpler scenarios • Not a suitable choice in a non-Windows or mixed environment
Potential Fit	Community Oriented, Informational, Multichannel Publishing
Unlikely Fit	Global Enterprise, Advanced Marketing Portal, Global/Enterprise Digital Workplace
Compare To	Sitecore, Ektron, Hannon Hill, Microsoft, EPiServer
O S	Windows and IIS in Management environment, can use Windows, Solaris, or Linux Delivery environment
Repository	File System — no database required
Client	Firefox, Internet Explorer, and Safari
App Platform	COM and .NET; Optionally Java in delivery tier
Licensing	Server-based option – Median deal size is about \$50K, including first-year support. Managed hosted option – \$1,800 to \$3,900 per month, depending on number of system users. Cartella collaboration system is \$18K for 30K users

Specsheet	Ingeniux: Content Management System 7.5 Summary
Ownership	Privately held

Summary

Ingeniux is a US (Seattle-based) company with close to 40 employees. Ingeniux CMS is a mature, predominantly Microsoft-oriented product built atop .NET 3.5 and XML-related technologies.

Ingeniux makes heavy use of XML. Some licensees (e.g., those with contributors that are accustomed to using XML-based structured authoring tools) actually favor Ingeniux’s straightforward, no-nonsense, form-oriented approach to content creation. However, you need to recognize that getting the most out of the product requires that someone within your organization must have experience with XML Schemas (XSD), XSLT, and other XML technologies.

Ingeniux CMS is a potential fit for customers who have structured content and a need to reuse or repurpose content in different formats. It is not ideal for enterprises needing to offer substantial business-user control over site interactivity, such as highly dynamic, B2C environments, or complex digital marketing activities.

Ingeniux offers a hosted version called Ingeniux OnDemand, which the company says now accounts for more than half of its business. It is a hosted, virtual server offering, but it is technically different from a Software as a service (SaaS) offering from vendors like CrownPeak and Upland Software. It is a “managed service.” That’s a subtle — but important — differentiation. While you benefit from the potential of more advanced customizations and the ability to implement your own security, you don’t have certain benefits that pure-play SaaS vendors easily provide (like automatic rollout of new features).

The company offers a collaboration application, “Cartella,” which combines an interesting set of document management, digital asset management, and social software capabilities — at an added cost of \$18K. Note that this is a separate codebase.

Scenario Fits	
Simpler Site	
Informational	
Microsites & Landing	
Mid-Range	
Basic Digital Marketing	
Mobile Site	
Community Oriented	
Complex Site	
Advanced Marketing Portal	
Global Enterprise	
Multichannel Publishing	
Ultra-Large Single	
Digital Workplace	
SMB/Departmental	
Global/Enterprise	

Introduction

This is the Microsoft CMS product that “almost was.” Ingeniux was founded by a group of Microsoft émigrés in Seattle who had previously — while still working at MSNBC in 1998 — prototyped a content management server based on XML technologies. Microsoft elected not to pursue product development on it, and subsequently went its own way.

Since then, Ingeniux has steadily matured its CMS product, going through seven major versions. Product development has recently picked up after slowing down for a couple of years when Ingeniux put its R&D resources into Cartella, a separate collaboration product.

Meanwhile, the Ingeniux customer profile has morphed somewhat. Originally, the company’s wins tended to be in regional media outlets. Then — thanks to a small handful of local partners and close attention to the university sector — the education market became a mainstay of sorts (which it still is). More recently, some trade organizations and financial services organizations also have gravitated to the product.

The overwhelming majority of the company’s business comes from North America. Ingeniux established reseller relationships in the UK and Spain, but they seem to have bore little fruit.

A business bet that has paid off is Ingeniux’s decision to emphasize its managed hosting offerings. The company claims that more than half of its business now comes from what Ingeniux sales staff refers to as “CMS by the Slice.” Ingeniux provides this service from eight data centers. It’s worth noting that a hosted software instance can be moved on-premise, if you decide you want to manage the datacenter aspects yourself.

After a long wait, version 7.0 was released at the end of 2010, which was quickly followed by version 7.5 in early 2011. Most notable of the new features include a new in-context editing environment, the ability to create workflows visually, a single unified client, and an interface that is available in multiple languages.

Technology

Technical Administration and Security

Architecturally, the CMS product is a decoupled system, with two main tiers:

- **Content Development Tier**
 This is the core content management layer and consists of the CMS and the repository. Your editorial and admin users use this to manage content or to develop the site.
- **Content Deployment Tier**
 This is a run-time delivery or presentation layer for end users. It could be a plain web server or Ingeniux’s dynamic site server.

The CMS is a Windows 2003/2008 application that uses .NET, IIS, and other Microsoft technologies. It uses a proprietary XML repository to store content. The repository is embedded as part of the application and uses the same framework. It provides storage as well as other content management functionality such as check-in/out, roll back, and workflows. The repository is stored directly on the file system. This kind of repository mechanism is useful in scenarios that require granular content reuse and decoupled environments. It also saves you from managing separate databases.

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

However, a proprietary repository may require specific skills, whereas a database (particularly an RDBMS) has well-established best practices and tuning guidelines. With this proprietary repository, you cannot take advantage of your database skills and other database utilities.

The delivery tier is independent of the CMS and uses any technology. You can deploy in any commonly used page technologies, including .NET, JSP, ColdFusion, or PHP. However, the complexity and code maintenance becomes problematic when mixing technologies.

You also have the option to use Ingeniux's own dynamic site server, which an IIS-based, .NET application with its own XML repository (it is an ISAPI extension installed on IIS web server). In this case, the XML content from CMS repository is replicated, and you can have dynamic delivery.

Ingeniux provides multiple APIs. There are .NET, Java, and COM versions available. Many features are also available as SOAP or RESTful Web Services.

All content is stored in the embedded XML repository and is published to different "publishing targets." You can do incremental publishing or do a full sync.

The Ingeniux CMS accommodates a number of possible setups. In the default fully decoupled, full-bake mode, content is rendered on the CMS server as final (X)HTML and replicated to a web server or file server where it can be consumed by site visitors. Ingeniux calls this "structured publishing," because the CMS deployment engine builds all of the physical directory structures on the web server and syncs all of the content automatically. There are no back-dependencies from web server to CMS since all deployed content is static.

Ingeniux also supports a frying model. In this model, content is replicated to a web server in a raw or semi-expanded format and rendered "just in time" by a server application when it is requested. Ingeniux provides a delivery-server (called Dynamic Site Server) to render content and aggregate data from databases, feeds, Web Services, and other sources dynamically. An additional RTA (Run-Time Authentication) module supports what amounts to pluggable authentication.

You can publish content using server-side technologies such as Java, PHP, or Cold Fusion.

Finally, you can also use a pull mechanism using a REST-based URL to query content directly from the CMS. This is not a good idea if you need separation between CMS and delivery. This also has implications on performance, as it puts a huge load on the CMS back-end. In most cases, back-end tuning requirements are different from front-end requirements.

After struggling a bit with its own lightweight deployment module, Ingeniux now bundles PeerSync (a third-party product) for deployment and replication. The package uses a non-transactional push model, which makes it easier to deploy content to heterogeneous environments, albeit with none of the quality-of-service guarantees that a transactional approach offers. Remember, you are just moving files — either XML to be transformed later on the fly, or fully rendered HTML — around in this setup.

Irrespective of the type of publishing you do (and you will probably use a combination of these approaches), remember that publishing is one of the most troublesome aspects of a decoupled content management and delivery setup. One user we talked to mentioned lack of

visibility into what goes on behind the scenes during the publishing process. You must explore this aspect to see if it works for you. Pay specific attention to incremental publishing and how the publishing operation affects overall performance.

Ingeniux CMS has something called the Cascading Authentication System to support a degree of flexibility in user authentication that's not commonly found at this level of the market. This module basically makes it possible to authenticate users via more than one identity store. Other than its own store, it supports Active Directory, LDAP, and a few custom stores. It's a delegated authentication model designed to appeal to large enterprise customers with elaborate identity-management infrastructures, who may need to support different authentication options for different audiences. Make sure that if you have specific requirements, the provider is supported. If not, you will have to write your own custom provider.

Ingeniux stores authorization information in its own user store. It allows you to sync with Active Directory and LDAP at regular intervals. You can configure this interval, but make sure it is not too small or too big. Keeping it too small will affect performance; keeping it too big will lead to stale data.

The system supports groups rather than roles. Every user has to belong to a group in order to use the system. An administrator sets global permissions on groups through a dialog with lots of check boxes. It's possible to define as many custom groups as you want or need, and associate particular groups with particular nodes in the site tree. Also, when setting up workflows, the administrator can assign specific groups to specific tasks in the flow.

Although the system is somewhat tedious to set up initially, the system permissions are quite and allow fine control over a given group's capabilities. This granularity, in conjunction with the delegated authentication model enabled by cascading authentication, provides some interesting, identity-based personalization capabilities. For example, it would be relatively easy to configure the system to enable permissions to check in content assigned to others and reassign that content's ownership, without enabling full editorial or publishing rights.

In terms of reporting, the system offers a custom reports dialog where power users can enter XPath expressions to execute queries that bring back tabular results on things like all pages that are assigned to a particular user, or all pages marked for retirement on a certain date. The system does come with a few pre-canned XPath queries that can be used as the starting point for creating custom queries, but there's no getting around the fact that in order to get the most out of the reporting system, you have to know XPath — and be willing to debug your own XPath expressions. Not friendly.

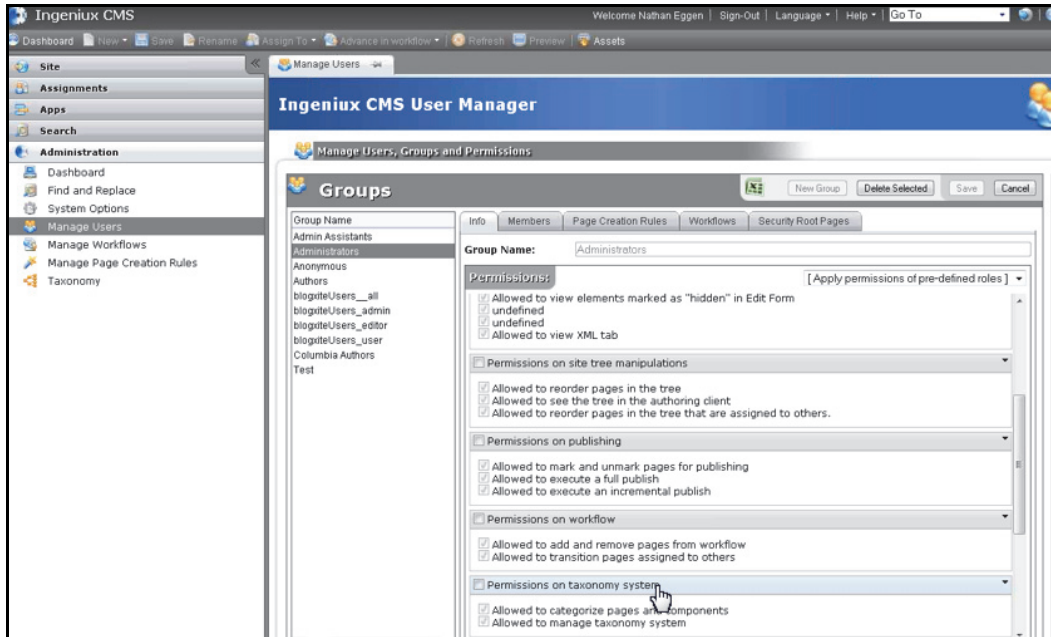


Figure 305. The User and Group Manager UI is now part of the same admin interface.

Development

Most developers use Visual Studio as the developer environment (although some also use open source Eclipse IDE). Developers create templates, content types, or extend workflows using Visual Studio.

Ingeniux supports COM and .NET APIs, with many interfaces also exposed through Web Services (SOAP and RESTful). To its credit, Ingeniux’s own R&D department uses its public APIs to build new features whenever possible rather than relying on low-level internals. (Ingeniux is not unique in doing this, of course, but smaller vendors are not famous for following this paradigm.) There is no support for CMIS and no plan to include it anytime soon.

Ingeniux has recently enhanced some of its XML-passing methods to use JSON (as an option) instead of XML, which some will see as a welcome simplification.

Note that Ingeniux stores and manages content as XML within a file system (rather than a database). You can optionally use a database for faster indexing and search of the content, though not for the management of that content. In practice, this “pure XML” approach means you need to take special care in modeling content structure up front, including schemas for validation. In this respect, the product differs from other file-based content management systems, like HP, that will enable you to manage unstructured (or at least, not as well-structured) files, such as HTML.

In Ingeniux CMS, you define content types as XML schemas, then associate XML elements for your content types and assign properties to them. Elements are first-class objects and can readily be linked and manipulated via visual UIs to develop powerful templates and site

architectures. This is powerful in reusing not just content but chunks (or specific fields like titles or abstracts) of content items.

You can use any XML editor, or use “Schema Designer” (Figure 306), accessed via Visual Studio, which saves XML schemas directly into the Ingeniux repository. Ingeniux says that trained business users can use it effectively, but it feels a bit more complex than competitors’ content-definition UIs. Anyone comfortable with tools like XML Spy will probably be comfortable with the Ingeniux Schema Designer. Others should get ready for a learning curve. Once you create a schema for a page or content item, the CMS creates content entry forms. The resulting content or page is then stored as XML and displayed via XSLT.

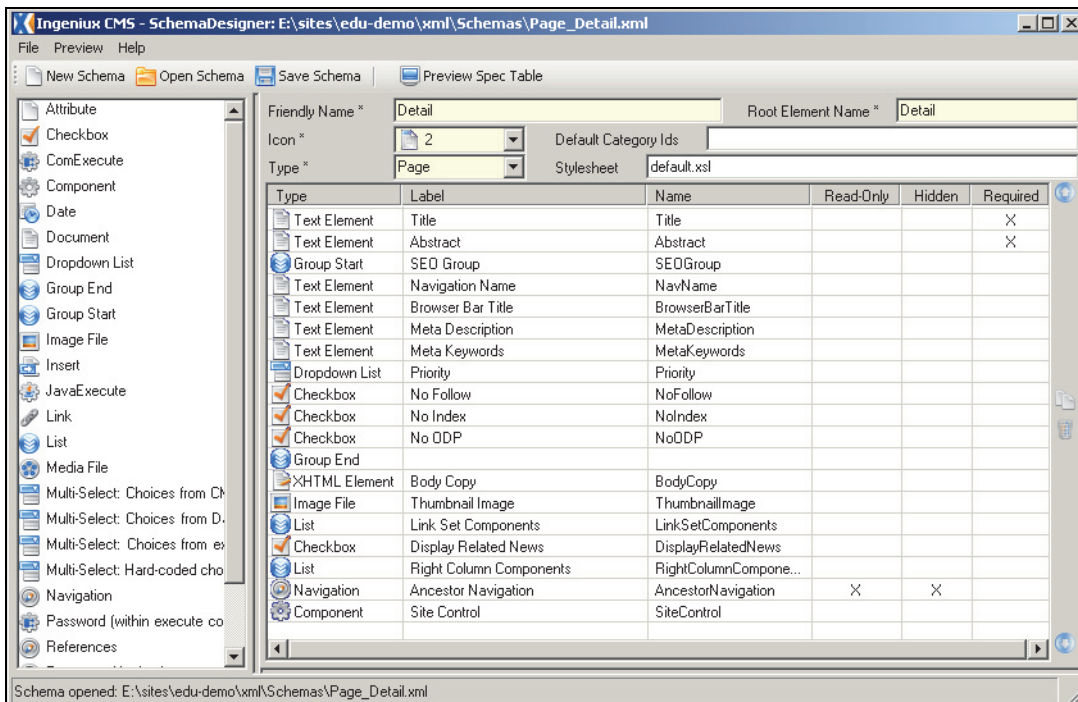


Figure 306. With Ingeniux, developers build content types using a special “Schema Designer” client. You can also use any standard XML IDE.

Power users can make changes to schemas through a point-and-click, browser-based interface. However, if you change a schema, it will only have an impact on the pages (or content items) that were created after that change. Pre-existing content items will not change, which could have some tricky content implications

Either client — thick or thin — you’re essentially defining elements and attributes through text fields and drop-down menus in dialogs, while the underlying logic generates a schema for you, saving you from having to hack together an .xsd file by hand (something no human being should ever be required to do).

Developers can create dynamic behaviors (e.g., automated navigation) by embedding XPath queries into component-builder forms (Figure 307).

The screenshot shows a web-based configuration interface. At the top are tabs for 'Edit', 'History', and 'XML'. Below is a search bar labeled 'Admissions'. The 'Number Of Stories' section has a dropdown menu set to '3'. The 'Category' section has a dropdown menu set to 'Admissions'. The 'DepartmentName' section has an empty dropdown menu. The 'News Navigation' section includes:

- Type: Children (dropdown)
- Order: (dropdown)
- Start Page: x6 (text input)
- Max Nodes: (text input)
- Max Depth: (text input)
- Query: (text input)

 Below the 'Exports' section is a table:

Name	Value	
Title	/*/Title	- +
Abstract	/*/Abstract	- +
Image	/*/Image[@Name = 'PageImage']/@File	- +
Category	/*/Category/text()	- +

Figure 307. Here, a developer is creating a dynamic index component using a browser-based form. Notice the use of XPath (in the fields at the bottom of the form) to allow things like Title, Abstract, Image, and Category to be determined dynamically.

XSL is a major enabling technology under the covers of Ingeniux CMS. It’s used to transform XML content to various output formats suitable for:

- Different browsing agents, such as PC browsers, cell phones, and so forth
- Different countries and languages
- Special clients, such as screen readers for the disabled
- Specific editions for downstream partners

Templates are essentially XSL style sheets that you write yourself and install separately in the system. That is, there are no browser-based template builders or configurators; nor does the system actively manage XSLT files the way it does XML objects. As a practical matter, it seems that Ingeniux’s professional services arm develops at least the first few templates for its customers as part of an initial consulting engagement. By default, each XML type is bound to an individual XSL display template, but authors can override that template manually.

To associate XSL with a particular type of story or page, you define “rules” in a rule-creator interface. A “rule” actually encapsulates not only the binding of an XSL style sheet to a page type, but also the binding of a schema to that type, and the assignment of rights regarding the rule (in other words, which users or groups get to use the rule). You can create new rules that inherit properties from parent rules, which saves time if you want to create many variants of a particular page type. When you’re done, a given story can be reused (using different rules) in

different situations. Rules don't automatically apply to individual elements (fragments) within a page, but depending on how you factor your pages, you can work around this.

Ingeniux can be configured to generate topical or dynamic RSS feeds, or more generally to output arbitrary XML for content interchange. Similarly, many Ingeniux clients import, manage, and publish external-source XML, such as data from an ERP system. (Concluded an architect at one customer site: "This technology gives me a warm and fuzzy feeling.")

Clearly, to take full advantage of the system's capabilities, you'll need to make sure that your developers are up to speed on XSL, Schema, and XPath. Otherwise, your implementation will likely be delayed. Even then, you may want to invest in training. Ingeniux offers an Introduction to Style sheets and Schemas course as well as a special package called Ingeniux 360, which includes 30 hours of direct access to Ingeniux developers (so your developers can get straight answers to the most technical of questions), and two days of classroom training for two of your developers, for a little under \$10K. If you intend to implement custom apps or non-trivial extensions to the product, you should budget accordingly.

Performance

From the contributor's point of view, there are no built-in performance bottlenecks except when bringing back large result sets from searches, but even then, you do see page results quickly (and can click on them); it's just that the facet hit counts and category nodes can take a long time to update. While they're populating, you can interact with everything on the page (thanks to AJAX), but if a result category has, say, several thousand items in it, you may wait 10 or 20 seconds — or longer — for the numbers to come back (Figure 308).

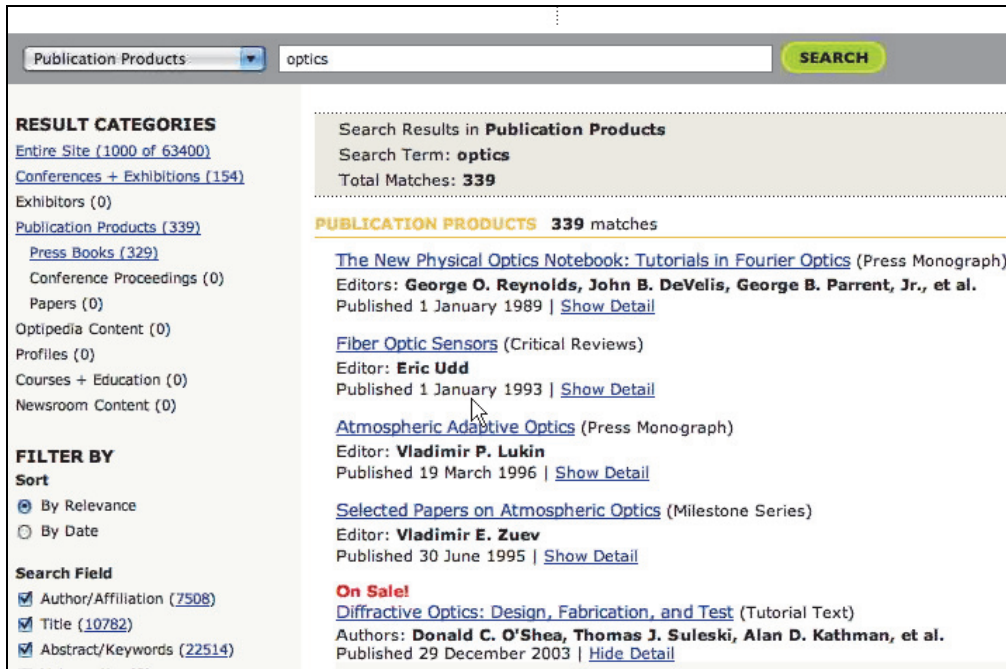


Figure 308. A search results page can bring back a lot of info. Note that some of the items on the left have hundreds or even thousands of items associated with them. It can take a while for the categories to populate.

Delivery performance depends on many factors, including page complexity and how much dynamic assembly is required (versus using static content), but at least this is a file-system based repository rather than a system in which assembling a page requires repeatedly issuing SQL queries to a database (à la Joomla!). The performance choke point you'll want to pay the most attention to is search performance. Plan to do some testing here.

Ingeniux reports that some of its customers take advantage of Content Delivery Networks for caching of images and other static resources. This provides for added redundancy and better performance under load, but can complicate matters if pages contain dynamic content. (This is an issue for CDNs generally, however; it's not just an Ingeniux issue.)

Content

Contributor Experience

Ingeniux has made several changes to the contributor interface over the years, including the addition of a single cross-browser dashboard interface for all users. The interface is appropriately permissioned, which means that it only shows relevant functionality based on user roles. As an administrator, you can disable functionality for certain users. For example, you can remove the ability to upload Flash files when using the rich text editor.

The left panel has a tree that enables you to see the site structure and navigate through pages and content. It shows all content; if you are a content-heavy site (such as a news site), showing all news items in the tree structure could have both performance and usability implications. Invest appropriately in structuring your site plan, and test to ensure it meets your requirements.

All of the functionality displays in the right pane. The interface is functional and efficient. It is built using JQuery and AJAX, and it supports drag-and-drop functionality.

Fortunately, there is only one interface, unlike previous versions, which had multiple interfaces for contributors and admin users. The advantages of a single interface are that you will have less training and lower management costs.

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

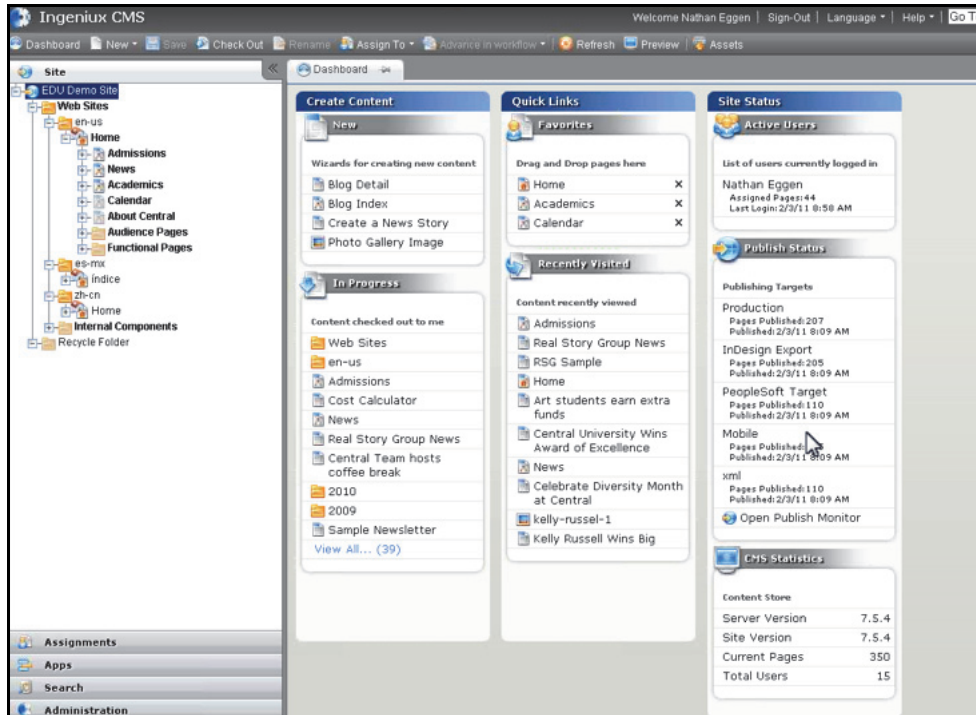


Figure 309. The dashboard is an AJAX-based interface for admin and contributor activities.

However, the disadvantage of having a single interface is that it inevitably becomes a crowded interface. Make sure that you enable very specific functionality based on user roles to be part of dashboards.

Because Ingeniux CMS is sold to customers in higher education and government, a fair amount of attention has been devoted to making the interfaces highly accessible (viz., Section 508 and WAI compliant), and since the system enforces schemas and is designed to output XHTML, the potential for creating accessible Web content is there; it's up to you to make sure your own in-house processes adhere to best practices around accessibility.

Interfaces are internationalized and ship in multiple languages. Out of the box, it comes in English, French, German, Italian, and Chinese.

Contributing Content

The WYSIWYG editor is a heavily customized version of TinyMCE. It enables you to strip outsource formatting or do a text-only paste. You can enable or disable different functionalities, based on user roles.

The content entry interface has multiple tabs. You can use a forms-based interface to enter content and click the XML tab to preview content items in XML. You can view analytics data and history.

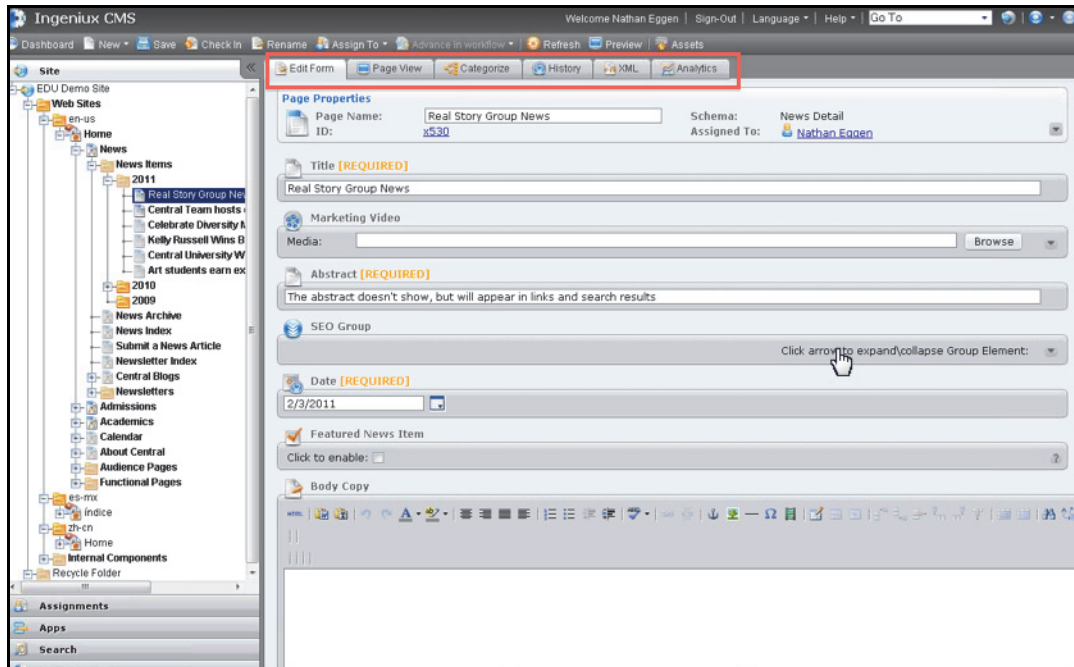


Figure 310. A tabbed user interface enables you to enter content using a forms-based interface. You can click on other tabs to see the underlying XML or to view other properties.

Version 7.0 began supporting in-context editing (accessed via the Page View tab). In fact, Ingeniux has taken a step further than many other vendors have by supporting structured in-line editing. This essentially means that instead of editing a big blob of HTML in-line, you can edit every single item of content (such as title or date field); it's powerful. However, the fact that this comes as part of the overall user interface and a user needs to login to the CMS interface could be a dampener for occasional business users. You should do your own usability testing, since we found random script errors and the in-context interface becomes clunky when pages have more content. This is primarily because there is less real estate available for editing, since the other parts occupy a lot of screen space.

As you'd expect of an XML-based system, content reuse is a major theme in Ingeniux CMS. Although content is mostly managed according to a page-based model, Ingeniux also supports a component model in which sub-page level chunks of content can be mixed and matched to form new documents. A "component," in Ingeniux, is an XML document without a style sheet or presentation. Granular down to the element level, these XML fragments can be either added to a page or assembled to generate larger documents. Thus, if your company is constantly creating bids, quotations, RFPs, procurement documents, proposals, contracts, or other documents in which pieces are often reused, it's possible to warehouse appropriate XML fragments that can be assembled into a final document.

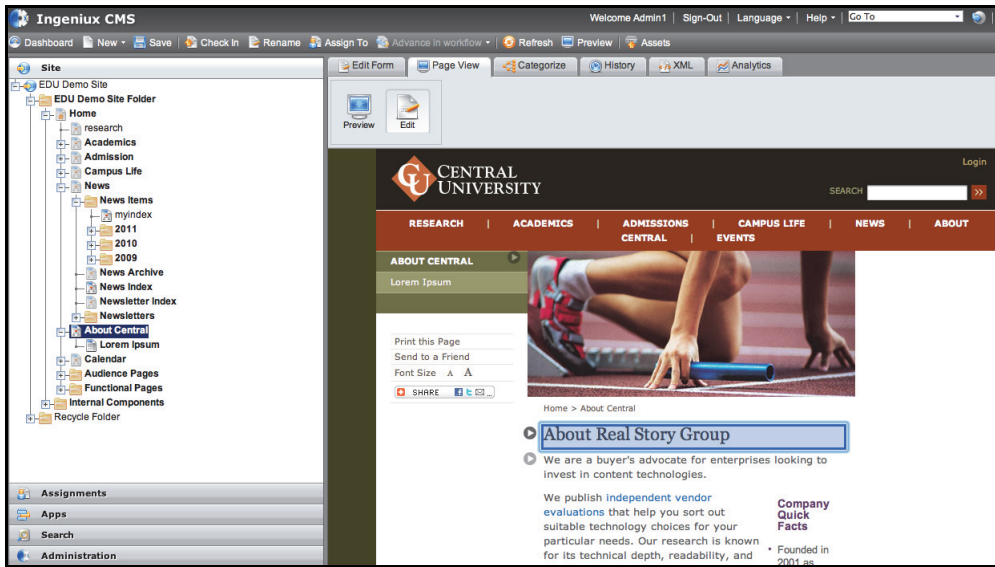


Figure 311. You can do structured, in-context editing in version 7 and above.

Note that Ingeniux CMS expects to deal in XHTML (not HTML) web content, and as a result the system is rather fussy about things like tags that aren't supported in XHTML (tags that *require* certain attributes in XHTML but not in HTML, and so forth). Accordingly, content is passed through HTMLTidy on the server, to clean up dirty syntax. (It will not insert semicolons into your JavaScript, however.) If you decide to bring text created in Word into the Ingeniux editing environment, a SmartPaste feature allows you to select from several “cleansing” commands to ensure that spurious Word markup is removed from text as it’s pasted in place. If you’re dealing in fairly small chunks of text for most page elements (which you should be), you’ll run into few problems. However, don’t expect to cut and paste large sections of non-XHTML compliant markup into the editor without someone — HTMLTidy, or you — doing some cleanup.

The system accommodates content repurposing relatively well and includes a Dynamic Preview mode for quickly seeing what various page variants look like. Using a framed interface (Figure 312), authors and developers can preview a page in different versions by applying various XSL transformations to the same content to simulate output to another site or client (handheld device versus PC, for example), as well as preview content in a development environment alongside other content still in progress. Note that you can also pass query strings, form elements, and cookies through this interface to simulate complex runtime behaviors, such as would occur with personalization. Of course, this assumes you have already created all the various XSLT (and other) files you need.

The system will warn a user if she tries to delete a content item that is reused by other content items, but will not include a report of all the affected objects and attendant dependencies. (On the plus side, restoring an item from the Recycle Bin automatically restores former dependencies.) The dashboard plugin system can be used to create detailed views of the dependency information for a content item, but this requires some customization.



Figure 312. Using “Dynamic Preview,” authors can preview content in multiple formats using a handy configurator at the top of the preview page. Note that in this example, the preview is of an .aspx version of a page.

Ingeniux has decent tagging and classification services. You can use these categories for “placeless content,” i.e., when the same content must display under multiple sections of the site. If a category is associated with an external taxonomy system, when you click on it, it shows all of the pages associated with it. A category can be permissioned, which means that you can see content related to a category only if you have permissions to it. A category can inherit permissions from a parent item, and it supports synonyms. Overall, the categorization system is powerful.

Ingeniux CMS ships with some basic aggregation facilities, such as pre-built components for ingesting various types of external feeds, like RSS (Figure 313) and weather feeds. You can also embed database queries into dynamic elements, but this is neither simple nor intuitive. On the plus side, some customers have used the product’s SOAP API to ingest and manage external content in the Ingeniux repository, as well as display web application data through Ingeniux-managed templates.

The product’s Repository Services capabilities are a bit light. The latest version allows you to store multiple previous versions of content items, but you can only compare those versions in an editorial screen (and not as part of a workflow process). You can version the underlying

XML content, but not the XSL that drives business and layout logic. This means you need to be very careful about applying an external configuration management tool (like Subversion) to your code. The product's asset management capabilities are comparatively light, but have grown somewhat in version 6, where it's possible to scale and resize images on the fly, create thumbnails, and apply metadata.

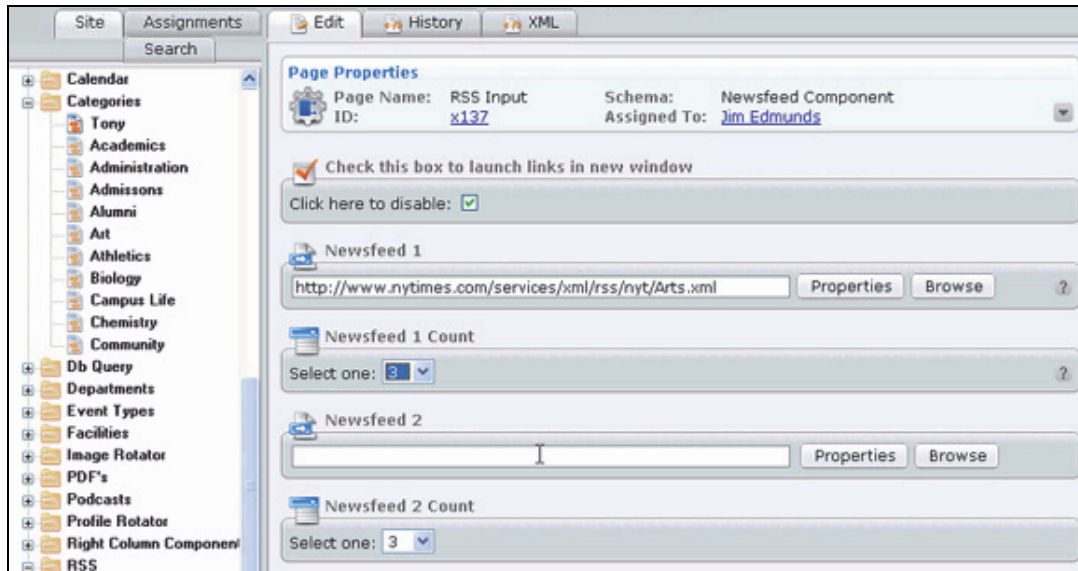


Figure 313. Editors can identify and configure RSS feed ingests.

For repository search, Ingeniux defaults to Lucene, but you can use a third-party search engine if you prefer. The search experience in version 7.5 has been improved, and the integrated Lucene search now supports element-level search. It is also more scalable compared to previous versions. However, Lucene (we review Lucene separately in our Search report) itself requires effort if you need any customization or tuning. Test these aspects well.

Ingeniux supports basic asset management capabilities. You can upload multiple images and do basic operations such as resizing, cropping, rotating, or automatically creating thumbnails. It offers integration with SharePoint if you prefer to manage your documents outside of the Ingeniux CMS.

Content Lifecycle

While not terribly sophisticated, the simple state-transition workflow system covers the basics. Developing and modifying workflow types is relatively simple, and you can create new workflows using a browser interface. This enables relatively non-techie users to create or manipulate workflows. If you need to add something to the workflow that is not available (such as integrating with a records management system), you will have to write a custom action using Visual Studio. While you can trigger all sorts of events easily, the end-user interfaces for work queues remain a bit thin. Moreover, the system suffers from some of the same shortcomings as other state-transition engines, inasmuch as it can be difficult to edit an existing content item without taking it offline.

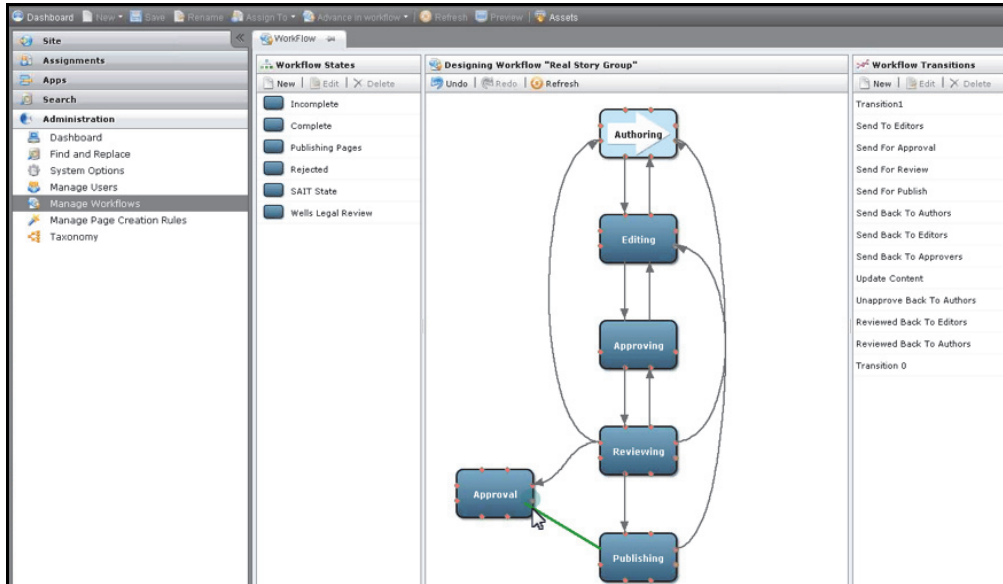


Figure 314. You can create new workflows using a visual, browser-based interface.

As you might expect from a product that sells primarily in the US, Ingeniux’s globalization capabilities are fairly weak. Its state-transition workflow does not support the kind of parallel processing that is frequently required in translation scenarios. Some clients have used a combination of version control, classification, and workflow to effect some degree of collaborative localization, but it’s more in the nature of a workaround rather than a native feature of the product.

Experience

Publishing

Ingeniux CMS ships with an XSL-FO (Formatting Object) processor that enables publishing PDF documents from XML content. Here again, you need to create your own stylesheets, although the product ships with some example code that will generate a PDF from a book content item, complete with a table of contents. However, it takes significant XSL expertise to realize the potential of this system. Some Ingeniux customers also push XML directly into InDesign and Quark, although the product does not come with any special tools to help you here.

Ingeniux CMS has the ability to generate SEO-friendly URLs natively. You can manage URL extensions, canonical URLs, and redirects from the admin interface.

Digital Marketing

The product supports runtime personalization (assuming you are “frying” your pages rather than baking them). However, to get the most out of the system, you’ll need XSL skills.

For analytics, Ingeniux provides integration with Google Analytics and Website Optimizer. The integration features a dashboard that enables marketers to set up Google Website Optimizer experiments (register various site pages, and get to results) right from within the Ingeniux CMS. Ingeniux was one of the first CMS vendors to become a Google technology partner on this. Upland Software has also taken this approach.

After focusing on pure page management, Ingeniux has paid more attention to micro-applications in recent releases. Out of the box, the product comes with basic polls, surveys, calendars, course catalogues, and student/professor profiles. Not all of them employ standards-compliant XHTML, but you can swap in your own schema for them and modify the XPath statements in the relevant XSLT.

To support a somewhat wider range of user generated content options, the company has begun promoting what it bills as a Social Content Management platform called Cartella (Figure 315). Cartella uses a fairly elaborate three-tier architecture comprised of Modules, Social Content Management Services, and Content Storage. The Modules include Document Management, Blogging, Wiki, Video (playback and transcoding), Image Gallery, and Social Services (Commenting, Ratings). For content management services, Cartella supports workflow, user management, sharing and permissions, notifications for reporting and outbound user engagement, tagging, and RSS syndication features.

Experience	
Publishing	
Standards Adherence	
Multichannel	
Mobile	
Digital Marketing	
Site & Campaign Analytics	
Testing & Optimization	
Segmentation & Personalization	
Social Media Integration	
Promotional Campaigns	
Community & UCG	
Workplace	
Collaboration & Networking	
Dashboard	
Ancillary	
Site Search	
Online Forms	
Module Ecosystem	

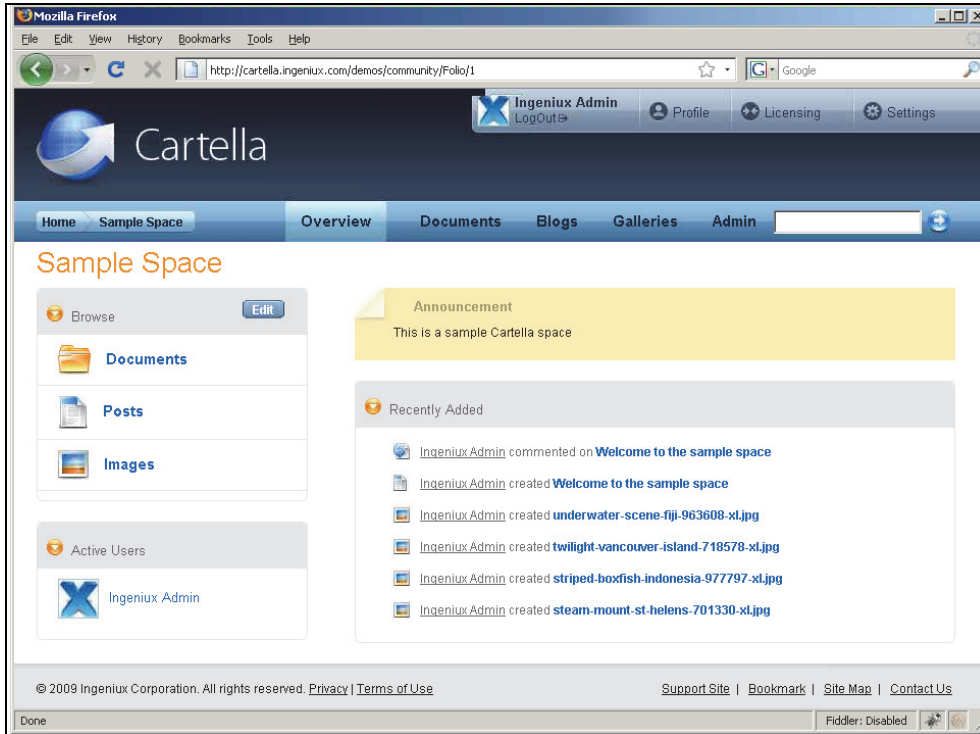


Figure 315. Cartella provides a blend of document management, DAM, and UGC features in an environment that supports user management and workflow.

With Cartella (a separate module costing \$18K), Ingeniux has taken the somewhat radical approach of declaring social content to be its own beast, deserving of its own management infrastructure, set apart from the conventional Web CMS. This is in stark contrast to the bolted on UGC approach taken by most CMS vendors, who treat social content as yet another bit stream that needs to be back-replicated into the CMS proper. It’s an interesting approach that bears watching.

Ancillary Services

For site search, you can use Ingeniux’s own integrated search engine (based on Lucene), or a third-party search engine. Fortunately, most commercial search engines — as well as Lucene — will take good advantage of the underlying XML file store, but test the search performance before deciding on your search engine.

Starting with 7.5, Ingeniux also provides an “App Manager,” to enable you to bring in external apps within the CMS.

Vendor Intangibles

Licenses for a single CMS server lists at US \$25,000 each, but with academic, non-profit, and government discounts, the median software deal size comes in around \$50K. However, the company’s most successful alternative licensing plan is the OnDemand option (which now accounts for the majority of Ingeniux’s signed deals), which puts it in direct competition with OmniUpdate in the North American academic market.

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

The on-premise version comes in three flavors:

- **OP Essential** – Covers 1 CMS, 1 Domain, and 5 concurrent users. Licensing costs \$25,000, plus \$4,500 annually for support and maintenance.
- **OP Professional** – Covers 2 servers, 15 concurrent users, 3 domains, and includes App Modules. It costs \$35,000 plus \$7,500 annually for support and maintenance
- **OP Enterprise** – Covers unlimited sites and users. Starts at \$60,000 plus \$12,000 annually for support and maintenance

The OnDemand version also comes in three flavors:

- **OD Essential** – Covers one website, with seat licenses for 5 concurrent system users. It is a shared application hosting (note: not multi-tenant) and starts at \$1,800 per month
- **OD Professional** – Covers 3 websites, 50 contributors and a dedicated VM. Starts at \$2,800 per month
- **OD Enterprise** – Covers unlimited users and websites. Starts at \$3,900 per month

As always, you’ll need to consider carefully the long-term staying power of any company this size — now around 40 employees. We think the Ingeniux technology is good enough — and the licensee base (built out slowly over a period of ten years) solid (and loyal) enough — that they would be a takeover target long before becoming a casualty of an overcrowded market.

The company says it is pursuing service as a differentiator, providing 30 hours of support and a yearly on-site audit as part of every deal. Customers report relative ease in contacting real developers at Ingeniux when the need arises (common at small software companies). On the down side, unlike OmniUpdate, there is no real active community around Ingeniux. The company’s customer support forums are not particularly usable, and appear to be fairly quiet.

Ingeniux takes pride in its professional services and keeps a comparatively quite small stable of consulting partners; but on the whole, Ingeniux development skills are not widespread.

Conclusion

On the whole, Ingeniux offers a relatively straightforward (if technically aging), XSL-driven CMS. It can boast decent ease of use, suitable for correspondingly straightforward scenarios of the kind typically found in the education market. It lacks sophisticated site management or web experience management capabilities; but similarly, non-technical end users face only a

shallow learning curve — and there’s just one client interface (the Universal Client). For those who prefer a “lease” solution, Ingeniux offers a number of managed hosting alternatives.

Customers with a need for a feature-rich, extensible, collaboration-oriented platform will find an interesting set of capabilities in Cartella. However, note that Cartella is still relatively unproven, which means that customers are definitely wearing the “early-adopter” hat.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Upland Software: Clickability

www.uplandsoftware.com

Vendor at a Glance

Specsheet	Upland Software: Clickability
Geography	Primarily North America
What's New	<ul style="list-style-type: none"> • Single sign-on via SAML support • Integration with Demandbase for visitor profiling and targeting
Strengths	<ul style="list-style-type: none"> • Company traditionally targeted functionality for media sector • Only major SaaS vendor to tightly couple content production and delivery, easing deployment, preview, and dynamic delivery • Comparatively broad site-management capabilities and user-generated content services, out of the box • Large tag library for customization by a reasonably technical webmaster, plus real development and staging environments • Most customers find the support team to be responsive and capable • Company's consulting team also held in very high regard • Impressive customer community has emerged for peer support • Redundant data centers offer greater reliability and continuity
Weaknesses	<ul style="list-style-type: none"> • In this coupled SaaS CMS, you will want to monitor performance and cost carefully; it's architecturally too coupled for some large enterprises • Comparatively burdensome integration and extension capabilities • Interactive applications lag in comparison to best-of-breed alternatives • Weak content lifecycle management • Overall approach is page oriented, not component oriented • Extremely limited digital asset management services • Poor user and group management capabilities • English-only UI — product interface is not internationalized
Potential Fit	Informational Sites, Basic Digital Marketing, Microsites & Landing Pages
Unlikely Fit	Digital Workplace, Global Enterprise, Multichannel Publishing
Compare To	CrownPeak, Ektron, Escenic, Sitecore, CCI, OmniUpdate, Drupal
OS	N/A
Repository	N/A
Client	Browser: IE and Firefox
App Platform	Hosted — tag library and APIs for developers
Licensing	Based on users, websites, and volume of visitor traffic. Annual subscription fee ranges from \$50K to more than \$200K
Ownership	Privately held

Summary

Upland Software/Clickability’s Platform — previously known as Limelight Networks’ Dynamic Site Platform, and before that as cmPublish — provides both a SaaS-based Web CMS, as well as website hosting services. The product initially fulfilled the needs of media sites, and thus has strong support for content prioritization and metadata. Upland Software Clickability continues to favor and do well in this industry vertical, offering a set of out-of-the-box customizations and functionality primarily targeted for media and publishing sites. Clickability appeals to organizations with limited to non-existent IT resources in a variety of industries.

Although Clickability can boast some larger customers, the platform remains best suited for mid-sized sites that don’t require extensive transactions or secure connections to internal systems. While historically it’s been a good fit for Basic Digital Marketing sites, or Marketing-oriented Microsites; it is increasingly used in the Ultra-Large Single Site and Multisite scenarios. Still, it is certainly not a good fit for Intranet or Global Enterprise scenarios, where you frequently need to integrate with sensitive internal systems.

You would not want to employ Clickability for intranet scenarios or public sites where you expect high- or spiked-traffic volumes due to potential performance issues and skyrocketing costs. Moreover, it’s not a good fit for multinational and multilingual publishing environments, or more generally, for non-North American customers — primarily because the interface comes in English only. However, several multinational customers use this product to publish out multilingual sites, evidently content with the authoring environment in one language only.

Scenario Fits	
Simpler Site	
Informational	<input checked="" type="radio"/>
Microsites & Landing	<input checked="" type="radio"/>
Mid-Range	
Basic Digital Marketing	<input checked="" type="radio"/>
Mobile Site	<input checked="" type="radio"/>
Community Oriented	<input checked="" type="radio"/>
Complex Site	
Advanced Marketing Portal	<input type="radio"/>
Global Enterprise	<input type="radio"/>
Multichannel Publishing	<input type="radio"/>
Ultra-Large Single	<input type="radio"/>
Digital Workplace	
SMB/Departmental	<input type="radio"/>
Global/Enterprise	<input type="radio"/>

Introduction

Founded in 1999, Clickability began as a hosted interactive widget provider to major media sites. The company’s Web CMS offering came out in 2002, and since then, the offering saw traction in verticals other than media such as high tech and financial services. According to the vendor, they have close to 80+ customers, about 400 published websites, and deliver around half a billion pages per month.

In 2008, Clickability received \$8 million in venture capital funding, bringing the total VC investment to \$15.3 million, and heralding a period of very rapid staff growth that then reversed itself in 2009 when it started shrinking in size. In May 2011, Clickability was acquired by Limelight Networks — a provider of CDN services, as well as hosted video delivery platform (transcoding/encoding, streaming and HD video), mobile, and other cloud

services and infrastructures. We tend to be cautious in evaluating the future of the platform in this (Lime) light.

Today in terms of WCXM, the vendor offers several options in addition to the core CMS offering. The first one, Website Express, is a set of standard content types, templates, and additional tools that allow you to build websites by configuring and selecting from a set of pre-canned options. The second solution, Website Marketing Accelerator (WMA), was launched in June 2010 and is targeted at B2B websites.

Compared to its direct competitors, Clickability also focuses on social applications, and offers a bundled “Social Media” platform, which includes forums, polls, and managing User-Generated Content (UGC). It does not include a wiki, which perhaps speaks to the dearth of intranet implementations. Several enterprise managers we spoke to have so far resisted applying Clickability behind the firewall.

Technology

Technical Administration and Security

The product is a bit unusual, since Clickability typically hosts both the CMS and the actual website in a combination package. That is, unlike CrownPeak, Clickability handles the delivery environment as part of a tightly coupled solution. However, there are ways to display content via other delivery environments like portals using the APIs.

The impact of this approach is that if you need to manage your own web servers or develop a lot of custom delivery logic, Clickability might not be right for you. However, if you need straightforward web publishing and don’t want to manage a heavyweight portal platform, Clickability is worth considering.

Clickability follows a tightly coupled publishing model. Since the platform hosts your websites, it will “fry” pages at run time. Content items are stored in XML, but transformed into Java objects the first time they are called, which avoids the repeated overhead of costly XSL transformation. The Java objects generate the page; it’s a bit unorthodox, but it works. In general though, a fully dynamic system like this is more likely to appeal to news organizations and other outfits with fast changing content, as opposed to traditional corporate sites that publish more slowly and deliberately, and update the same content items with some regularity.

Note that Clickability can support separate Dev, Staging, and Production environments. These are logically (rather than physically) distinct. In any event, you can test before setting an item “live.”

Groups and role management are some of the most cumbersome exercises customers report doing in the CMS. The Platform has the ability to create hierarchies within accounts that can be used to divide sites into segments with distinct permissions. By applying hierarchical structure with permissions, you can align content access with logical divisions within your

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

enterprise. Although the set up is pretty straightforward, the problems begin with managing permissions, which is a multi-step process and often frustrating experience.

In some cases, customers report having to submit a ticket to change/modify permissions, which eats up their precious professional services hours (some support tasks are done by the vendor’s pro services team versus the traditional support model).

Day-to-day users and admins of Clickability have described permissioning as “kludgy” and “awkward.” The lack of usable LDAP/AD integration is not a helpful factor either. Single sign-on (SSO) capability was added as in 2011 for published websites, with additional support for the WCM application in 2012. Both are implemented using SAML (Security Assertion Markup Language). With that in mind, be prepared for potential complaints about forgetting and resetting passwords. However, once you implement the latest SSO technology, there’s a chance to lessen the frequency of calls complaining about an inability to log into the CMS. As for internal reporting, there’s very little. CrownPeak edges Clickability here.

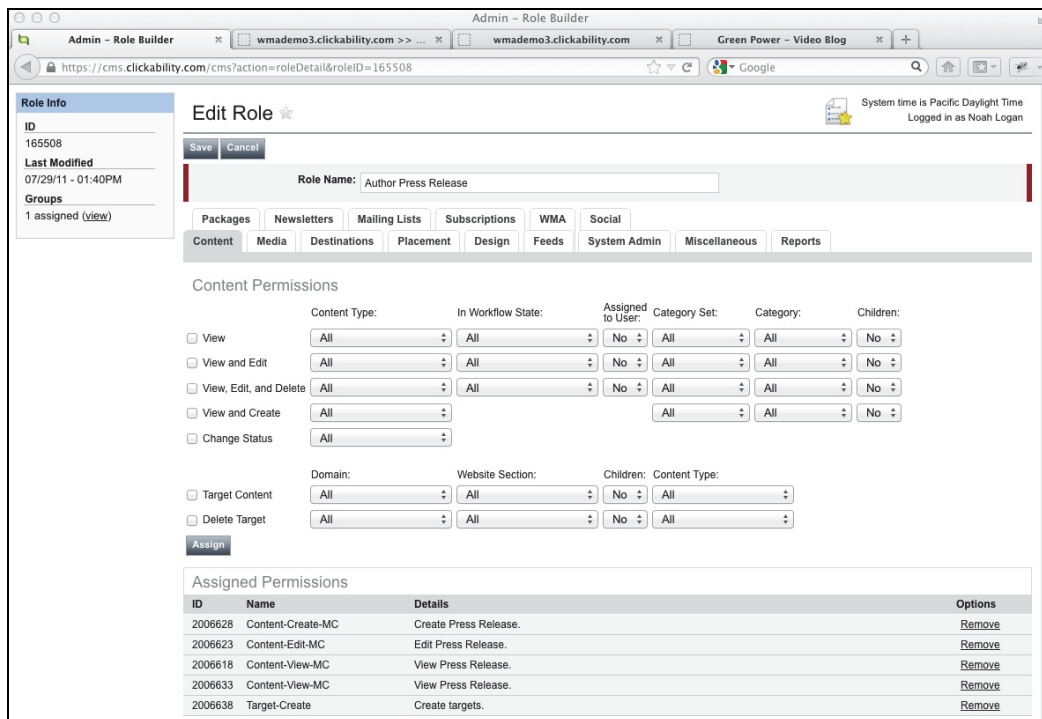


Figure 316. A glimpse into permissions settings based on user role, content type, workflow association, child-parent relationship, and a multitude of other variables.

Development

Clickability feels like more of a power-user product than a developer platform. The only expertise that’s required to build templates is that you learn the Clickability tag library, which happens to be reasonably broad.

The underlying templating engine is based on the Apache Velocity framework, which adds custom tags into HTML framework. You need a developer to build display templates through

a browser interface, using Clickability’s own tag libraries. Template elements are nicely nestable. Like CrownPeak, you use the same engine to create both content input and output templates. However, most customers rely on Clickability to create and modify them.

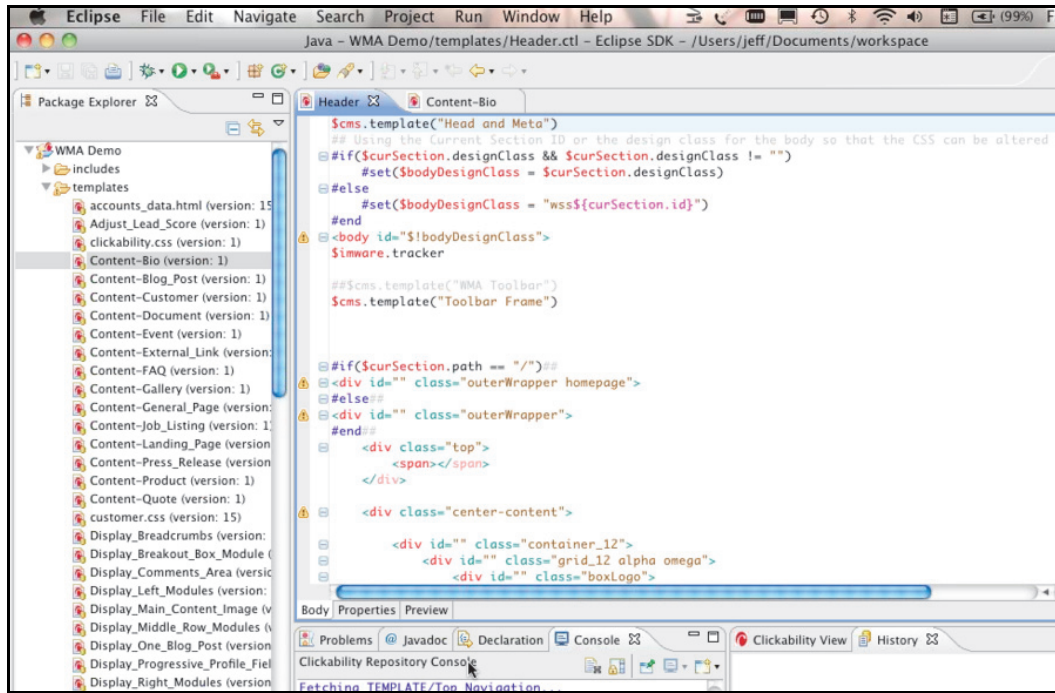


Figure 317. The templating engine is based on Apache Velocity and a proprietary tag library.

Not surprisingly, Clickability discovered that many developers were coding their templates outside of the system. For developers that need the ability to employ their favored IDE, Clickability has “Platform IDE,” an Eclipse plugin that allows developers to create and manage different artifacts directly from their desktops.

Some of Clickability’s larger clients are starting to experiment with the product’s APIs — Content API and Authentication API — to extend the platform. The Content API is used for extension and manipulation of content, and the Authentication API can be used for creating and editing visitor profile information, and for validating credentials. Some examples of how these APIs are utilized include publishing print content online, data mining and content enrichment, and integrations with third-party CRM and e-commerce systems. However, make sure that the depth of these integration capabilities suits your needs.

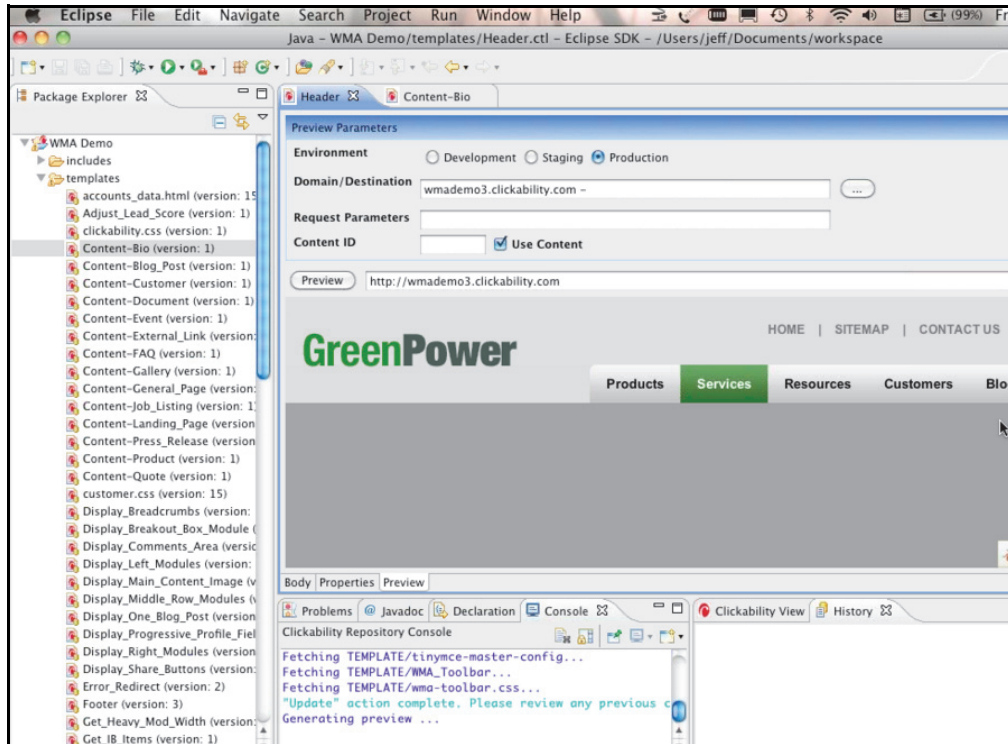


Figure 318. Template preview in Eclipse IDE.

The vendor claims that it ultimately wants to become a platform on which customers can build a variety of custom applications. To prepare the product for this possibility in the short term, Clickability has begun providing more access to its APIs. In any case, the benefits of the APIs will be highly dependent on your specific requirements; test extensively.

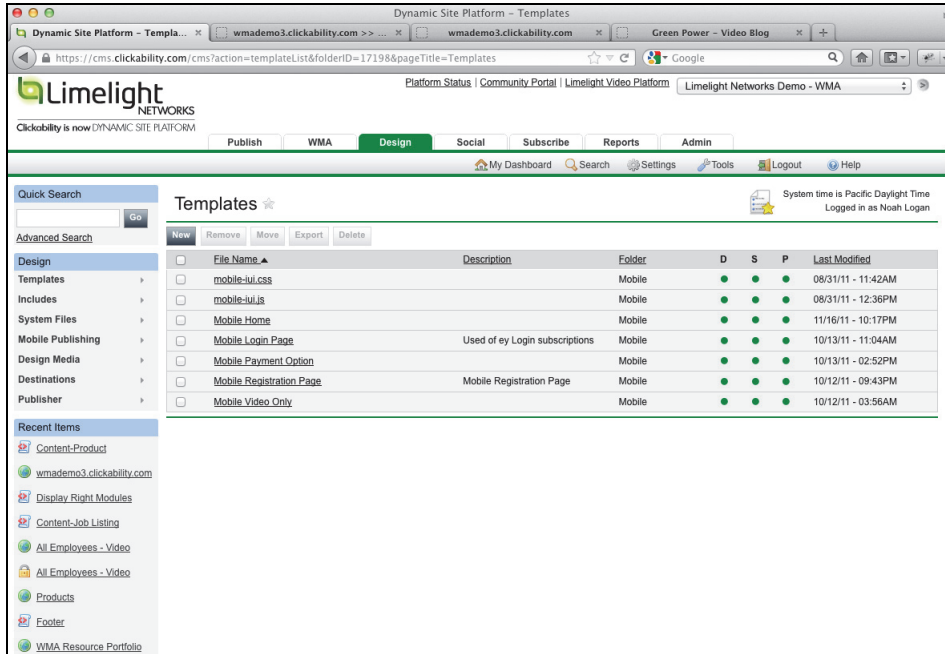


Figure 319. Browser-based template management view in DSP.

Unlike some other tools — both on-premise and hosted — Clickability has clear notions of Development, Staging, and Production environments, at least for templates and some other configurations (Figure 320). Since it is a coupled system, it can assign (in preview mode) different content to different stages. It’s a handy way to test a section or collection of pages that you want to push forward together.

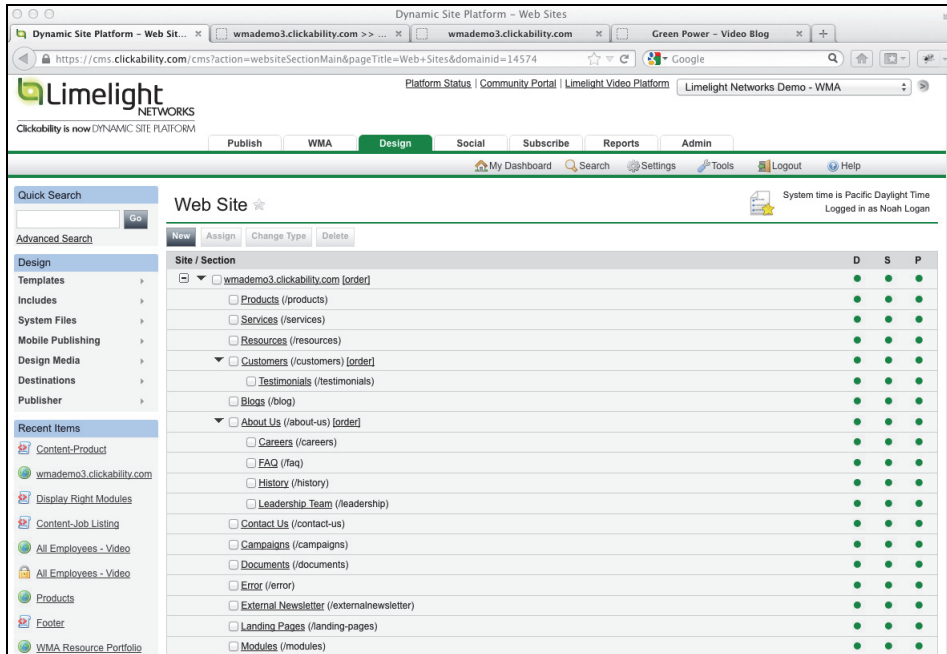


Figure 320. Administrators can see which content items are activated in the development, staging, or the production stage.

The product’s integration options remain relatively shallow. The major complaint we’ve heard from larger Clickability clients is the lack of ability to integrate with other enterprise-level systems securely and dynamically, such as ERP systems. The company claims to be increasingly aligning themselves with interoperability standards, although we haven’t seen evidence of this, yet.

On the plus side, the product also offers an excellent forms-based interface to build new content types. Customers report that it is easier to maintain looser content types in Clickability’s XML-based system, which enables them to apply formatting and relationships at different places in the content hierarchy.

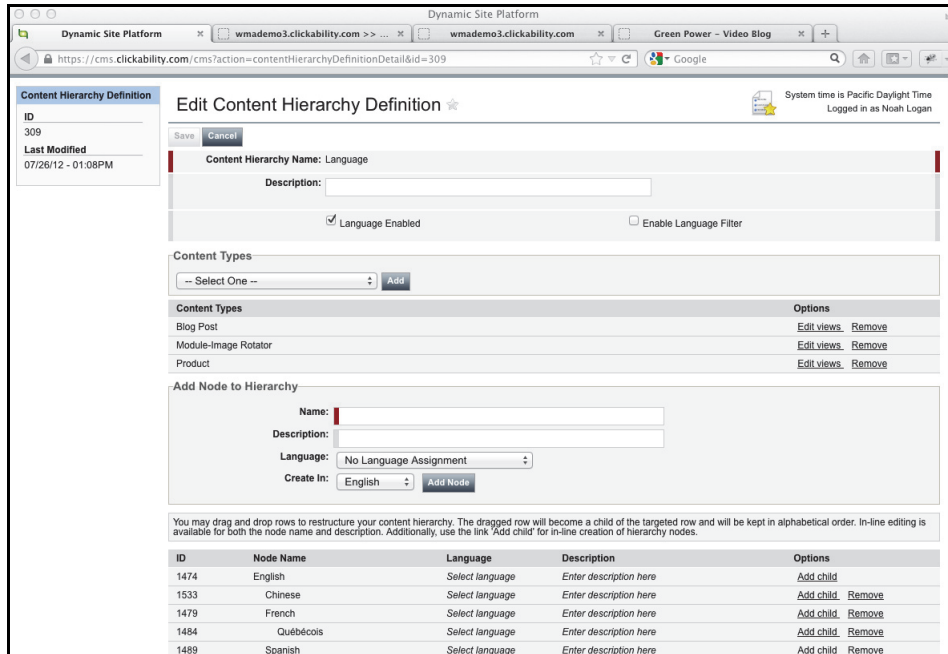


Figure 321. Assigning content types within the existing content hierarchy is an admin task in Clickability.

Content Connector is another developer tool that can be used for heavy-lifting integration scenarios — albeit in a limited set of circumstances — but you should be aware of all the complexities that lie behind this. In fact, several customers say that Content Connector’s innards are so complex that only the vendor’s own API developers can find a clear way through these muddy waters. If you do use Content Connector, depending on your implementation, your maintenance schedule may need to include (as much as weekly) touch-ups.

Just what is Content Connector? In essence, it’s an API framework around the SDK, a standalone Java web application, mainly focused on outside data import into the CMS, including XML feed ingest. The application makes abstract SOAP calls, focusing on the data and its proper formatting into the XML format. The SDK methods are then used to create content assets based on the XML ingested. If your use case is more complex than bare-bones ingestion and you have high volumes of data, you may need to use the SDK methods in combination with the Content API — which is more than a little bit of work here.

One thing your developers may not appreciate is poor code version control, which makes sizable deployments of new code releases rather painful. However, the vendor is coming up with little improvements here and there to appeal to the developers.

On published pages, your developers can attach debug=full as a parameter on the URL (e.g., <http://www.yoursite.com/?debug=full>) to see the order of templates used to build the page and how long it takes to generate each template.

One of the latest additions is developer diagnostic tooling. Loss of information while working on a SaaS platform due to connectivity and browser issues is common; this is one of the major weaknesses of the SaaS WCM model.

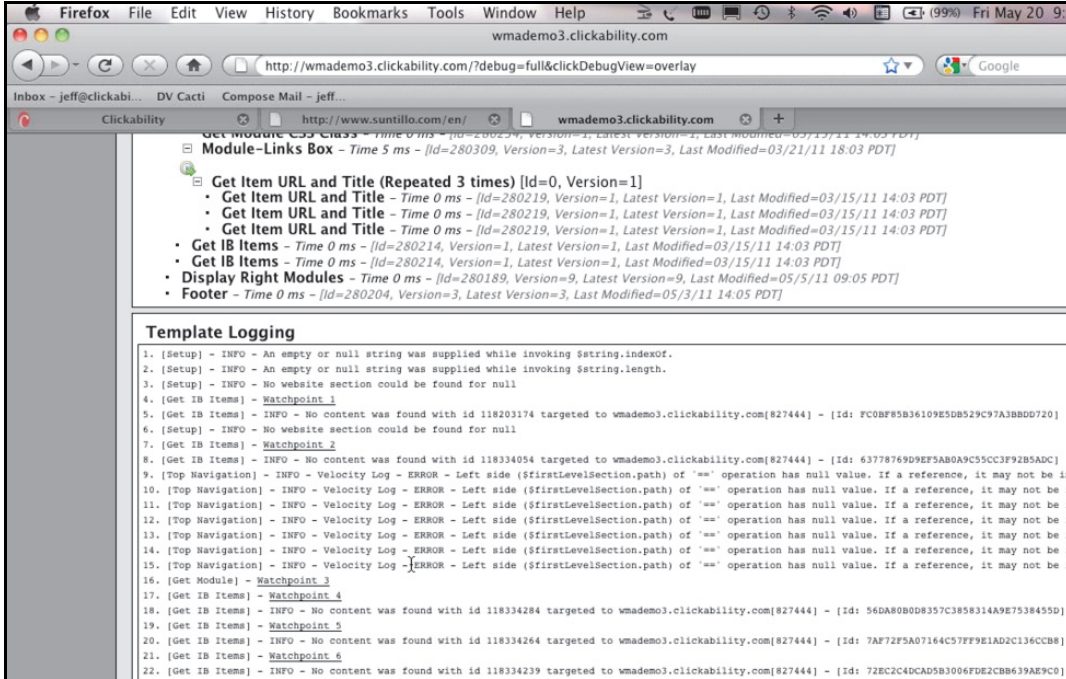


Figure 322. Debug mode and template logging shows template calls within a page, objects used to build the page, referenced objects, object counts, and other potentially useful information for developers.

Performance

Performance both for contributors as well as end users appears to be fast enough.

Clickability utilizes caching at different levels in its infrastructure for both performance as well as scalability. They use HTTP page caching using providers like Squid and Akamai, as well as object caching using Ehcache and Memcache. You can customize the behavior of these caches as well as disable one or more of cache types. However, caching is often tricky in general; make sure you carefully test features such as cache invalidation, setting different cache parameters, and handling caches in a clustered environment. Some customers report “funkiness” in caching behavior, which is admittedly rare, but tricky to track down when it does happen.

Clickability also has redundant data centers with automatic failover and real-time data replication. One of the customers we spoke to mentioned that their SLA had 99.9% availability and that Clickability had met that SLA consistently.

Content

Contributor Experience

While most customers seem to find the product easy to use and easy to train content producers, the look and feel of the product’s core features seem dated, especially compared to the company’s social media offerings. CCI has been working on streamlining the user interface, but refinements remain incomplete. Not all existing customers have moved to the new interface, but to be fair, that’s probably because they don’t want to train their editors again. In any case, test the interface well.

Content	
Contributor Experience	
Overall Usability	◐
UI Accessibility	◑
Contributing Content	
Authoring & Transformation	◐
Tagging & Taxonomy	◑
Content Reuse	◑
Media & Document Management	◑
Repository Services	◐
Content Lifecycle	
Workflow	◐
Globalization	◐
Archiving & Compliance	○



Figure 323. The latest interface for in-context editing in Website View, where editable and non-editable areas are clearly marked.

In general, Clickability supports IE and Firefox on the PC, and Firefox on the Mac — with Chrome working on both, albeit without official support. Unlike CrownPeak (which has given up trying to support the Firefox browser), Clickability has embraced Firefox, creating the Website View feature, which enables users to browse their site in their main browser window, while providing users with various views of analytic statistics in a sidebar. The company deprecated prior related features like the Insight Editor plugin in favor of functionalities called “Quick Links” and “Website View” for in-context editing, making it available in IE and Firefox. Moreover, only the following versions are supported: IE 8 (not lower) and Firefox 3.x. A good portion of functionality is missing in other browser versions.

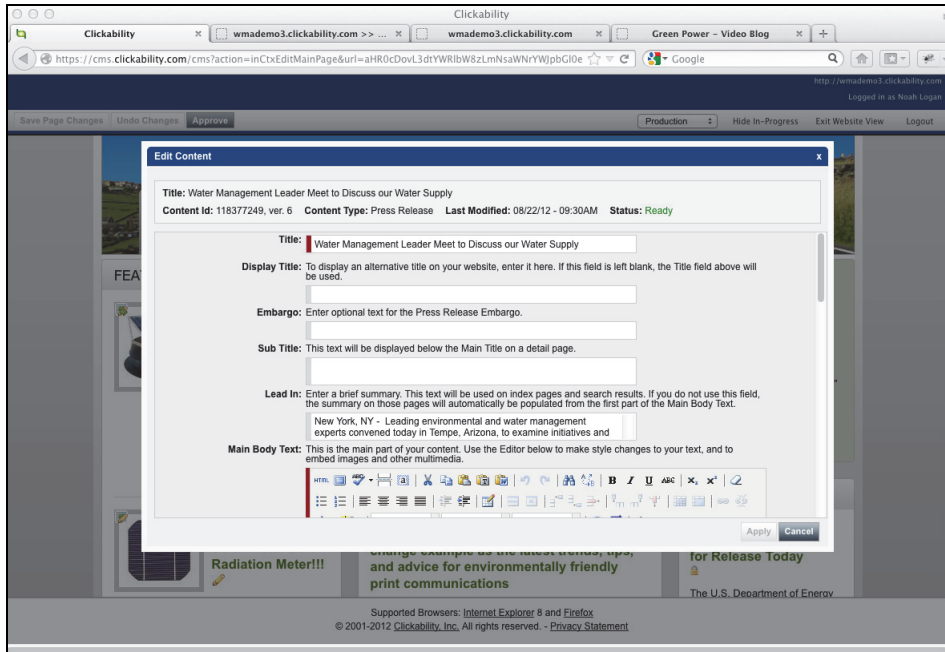


Figure 324. In-context editing in Clickability.

International enterprises should note that the interface is not internationalized; therefore it cannot be readily localized beyond English. In addition, international customers may need to employ certain kinds of magic to obtain tech support during non-Pacific time zone business hours.

Contributing Content

Clickability's authoring feature set is evolving. Overall, the product is geared toward news media organizations, and indeed, many of Clickability's content management customers are from that vertical. As an example of media-oriented functionality, Clickability enables editors to "order" lists of links on index pages in an ad hoc fashion (Figure 325) — a requirement typically found on news sites and almost nowhere else. Users have the ability to drag and drop articles into the list, rather than clicking arrows up and down to move items. Media customers also can easily ingest and republish external RSS feeds.

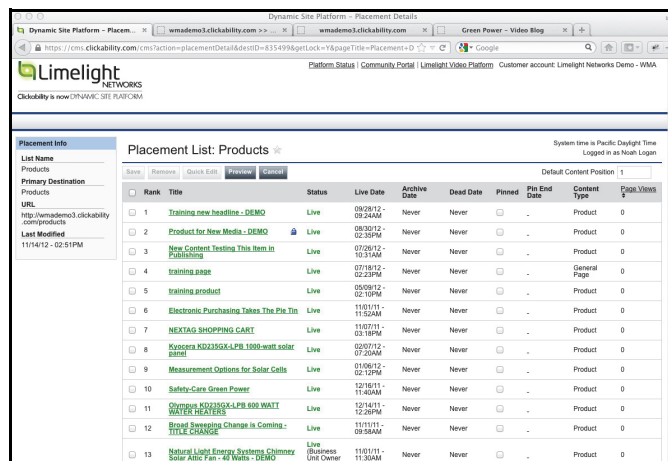


Figure 325. Ordering links manually via drag and drop.

Site “sections” also play an important role in the system, just as they would in a newspaper. When you create content, you physically assign it to a section of the site. Within each section, authors can choose from multiple display templates, determined by content type.

Clickability gives you a choice of using either TinyMCE or FCKEditor for having rich text control fields in content entry forms. Test the resulting code carefully; some customers have reported that they didn’t get what they thought they would, based on what they saw in a demo of the WYSIWYG editor. In particular, all-important Word cleanup appears inadequate.

Clickability has what it calls “Quick Edit” functionality, which enables users to edit multiple content items at once. While not all users will use this feature, it is particularly useful for publishers that need to make changes in multiple articles, all at the same time.

It’s relatively simple to assign metadata to content in Clickability. Clickability can store different sets of controlled vocabularies, using what it calls “category sets.” They can be managed globally, and you can assign those sets to specific content types. Here, Clickability straddles between some high-end packages that enable you to maintain a completely centralized taxonomy, and lower-end tools that only maintain a collection of site-wide keywords, at best.

Content contributors also can assign related items to any piece of content. This can be done on an article-by-article basis, or by creating customer-based views, but it’s still a manual process. While useful, this could become cumbersome for large newspapers that publish hundreds of articles a day.

For enterprises willing to pay extra to satisfy higher-volume needs, Clickability has partnered with an add-on service, Inform, to provide API-to-API content linking.

Like most systems, you can build out new pages or sections of a site and set a publish date for the new content to go “live,” but only in batches, not on an item-by-item basis, via the concept of Publishing Packages. However, customers have complained that there is no way to set a publication date on preexisting content items. Without this rather common functionality (in other systems), making global content changes to many areas of the site at the same time becomes a major hassle, especially when conducting site re-designs. Manually making these changes can be time consuming and difficult to coordinate proper QA processes. Again, Clickability seems optimized for sites that accumulate lots of new content, as opposed to those that primarily make updates to existing material.

Clickability’s repository services are reasonably good. The platform includes a handy “diff” check, which enables editors to compare versions of content and its metadata by highlighting the differences in color, and striking out the older content (Figure 327).

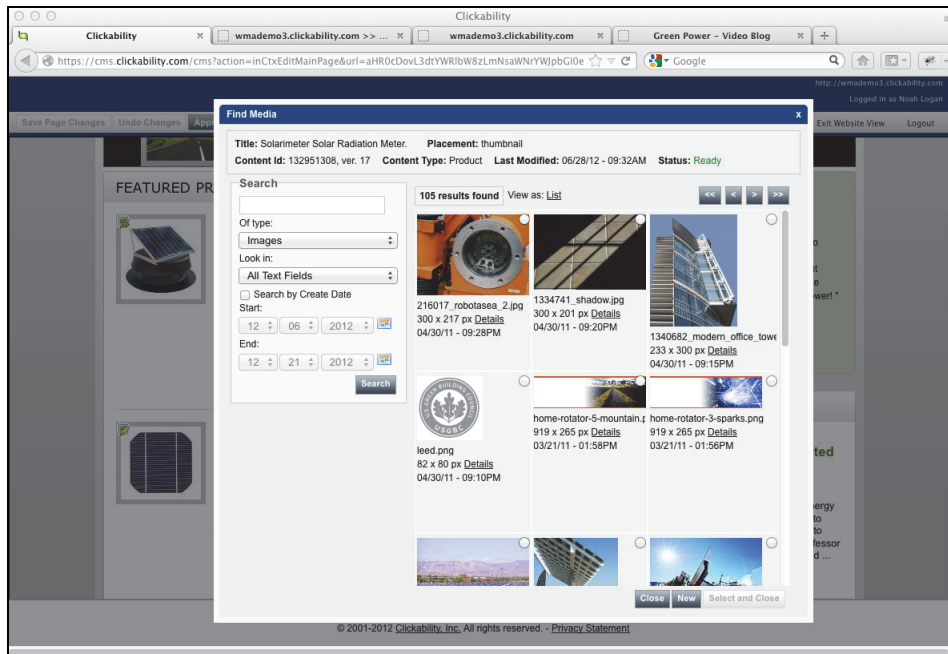


Figure 326. The interface for changing or replacing images stored within the CMS.

Clickability has an especially useful, fielded content-retrieval engine. Contributors can save advanced searches as personal “views” for repeated functions (e.g., show everything that was published that week, for example).

Digital and media asset management is comparatively weak. On the plus side, files such as PDFs and images can be tied to individual content items, or they can be uploaded singly or in bulk to a general storage area called the “Media Server.” However, management of these files is extremely limited.

Customers we spoke to expressed frustration over the fact that they had much less control over their files stored in the Media Server. There is no way of organizing these files in folders or hierarchies, no way to apply metadata, no version control, and no way to apply friendly URLs. As one customer stated, these files “are not treated like real content.” Customers are often forced to apply HTML wrappers to these files in order to replicate “real content” functionality.

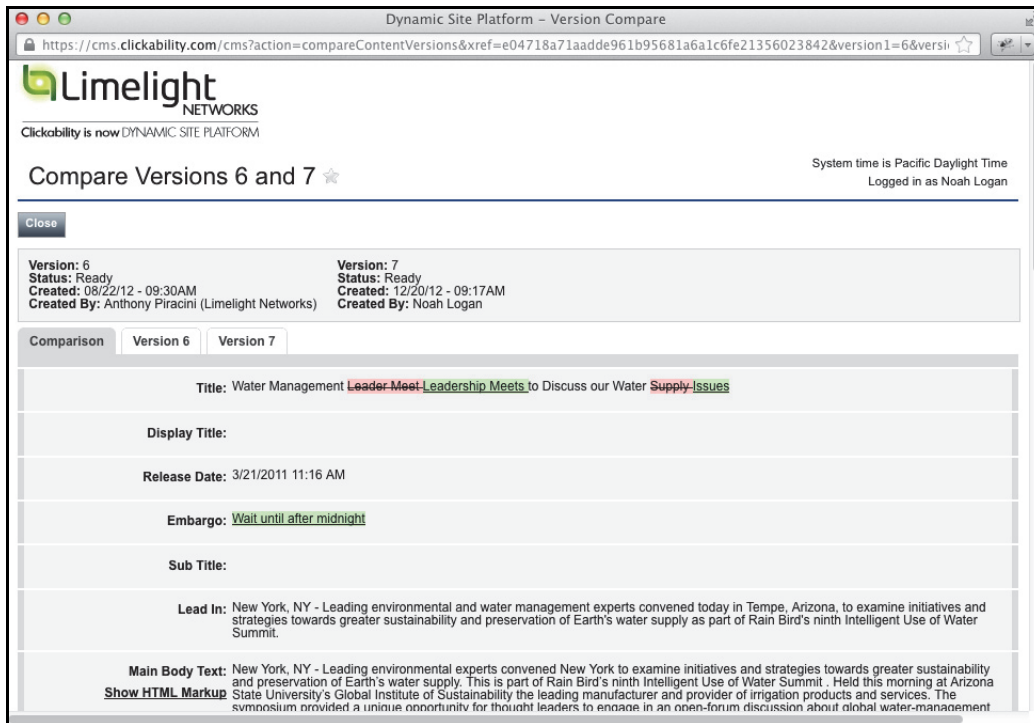


Figure 327. The compare feature enables users to see content differences between versions.

With Clickability, you can assign content to specific places in a site or sites. The system can reference all of the different places where an object lives, which is essential for this kind of re-purposing. On the whole, though, this is a page-based — not a component-based — system. Pages can live in different places, but you won't want to use Clickability to manage sites with fine-grained content models.

Content Lifecycle

The product's lifecycle services are rather thin. Out of the box, you can assign specific dates to three content states: "live," "archive," and "dead." This is better than some low-end systems that only have "live" and "expired," but it is important to be able to create custom states.

The product's workflow features are lighter than those of CrownPeak. Workflows are assigned to content type by default, but tasks *can* be assigned to groups, as well as to individuals. Clickability provides a browser-based workflow builder for identifying custom states and transitions. Workflow can only be linear; conditional or branched states are not supported. The product does provide a nice feature that enables you to preview or approve content through email, rather than logging into the system.

Unlike many packages, Clickability's CMS was originally designed to handle multiple sites under one owner, such as a broadcast network (one of Clickability's early client types). Clickability enables you to clone sites; through the interface, you can manage different domains and assign the same content to multiple sites. Site administrators (or their proxies) can manage roles, groups, and templates at either a single, domain, global, multisite level.

Enterprises with multilingual requirements have spawned modifications to the product, and Clickability has responded by adding new hierarchical content structuring capabilities that enterprises can use to create parent-child relationships between content items.

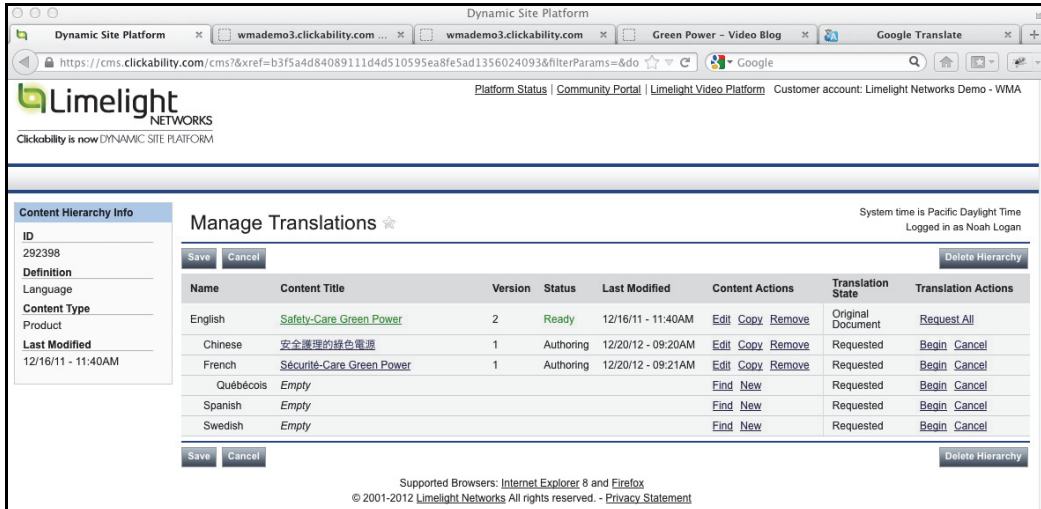


Figure 328. The content hierarchy displays any and all translations associated with a particular content item, including workflow status.

In addition, they've added RTL language support within the content entry interface, and the ability to tag content in the native language. However, you should not confuse this with true globalization services, where changes to source content automatically kick off translation workflows and subsequent reconciliations. (The company says there are workarounds using their API.) Clickability can support multiple languages for a US government news site, but not nearly as effectively for a multinational publishing operation for a global enterprise.

Experience

Publishing

By default, when the system assembles a page, it names it by number. Most customers override this; Clickability can automatically create a URL based on the title of the content (or you can manually override it). Clickability now includes a nifty interface that replicates many of the features of mod_apache to control various redirects — including vanity URLs.

For mobile publishing, in 2012, Limelight added “Dynamic Site Platform for Mobile,” which was an add-on module for DSP with some responsive design capabilities for templating and device detection. It’s a pretty basic offering at this stage, and it will require extra work to put it all together nicely.

Experience	
Publishing	
Standards Adherence	<input type="checkbox"/>
Multichannel	<input type="checkbox"/>
Mobile	<input type="checkbox"/>
Digital Marketing	
Site & Campaign Analytics	<input type="checkbox"/>
Testing & Optimization	<input type="checkbox"/>
Segmentation & Personalization	<input type="checkbox"/>
Social Media Integration	<input type="checkbox"/>
Promotional Campaigns	<input type="checkbox"/>
Community & UCG	<input type="checkbox"/>
Workplace	
Collaboration & Networking	<input type="checkbox"/>
Dashboard	<input type="checkbox"/>
Ancillary	
Site Search	<input type="checkbox"/>
Online Forms	<input type="checkbox"/>
Module Ecosystem	<input type="checkbox"/>

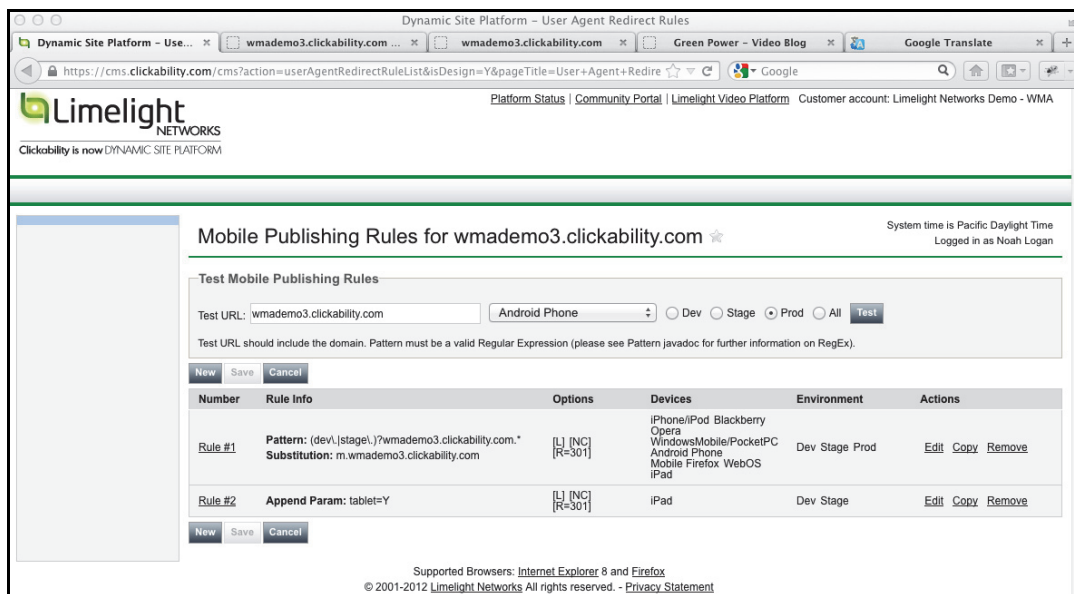


Figure 329. Setting up user-agent redirect rules for various mobile device types is rather rudimentary, but it’s more extensive than what some of the competition provides for mobile content management.

Digital Marketing

Runtime page assembly should theoretically bring personalization opportunities, but Clickability is still feeling its way here. To be sure, authors can push content items to pre-set visitor profiles. The company concedes that location services and detection, for example, are not out of the box and require coding.

Most location services and detection, for example, are not out of the box and require coding. In addition to its built-in personalization capabilities, the vendor recently added an integration with Demandbase’s customer profiling tool (used also by CrownPeak and Ektron), in attempts to identify B2B users from reverse IP lookups and to offer personalized content on industry sectors and geography, etc. This should help profiling to be more accurately targeted against your segments.

In June 2010, the vendor released a new add-on product called Website Marketing Acceleration (WMA — Figure 330). WMA is targeted for B2B marketers and helps to streamline the sales process. The module is a separately licensed offering that runs on top of your WCM instance.

At a high level, WMA enables you to gather information — both anonymously and based on user input. User input is collected over time (called “Progressive Profiling”), and you can rate or qualify a visitor based on specified parameters. All of this information is collected over a period of time, starting from when the user was anonymous, to when the user registered and provided some information, and based on user behavior. Based on all of the collected information, you can segment visitors and then target content specifically to them.

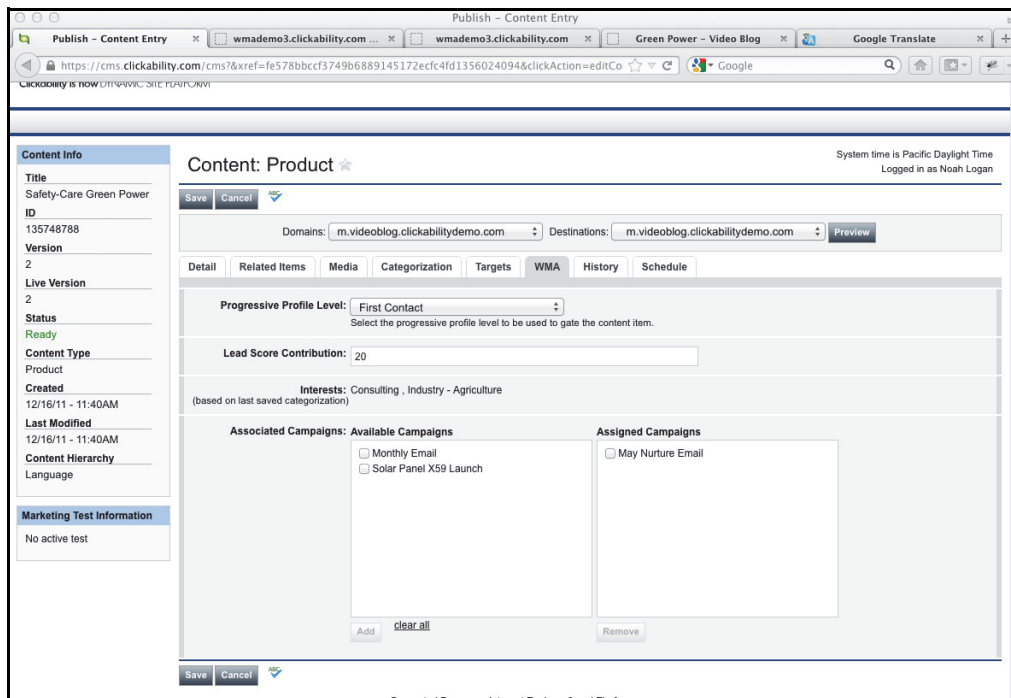


Figure 330. The WMA module enables you to target content based on progressive profiling. It can also be integrated with salesforce.com to get lead data.

WMA got off to a good start in the arena of web experience management, but remember that even though you can personalize and target content, it is not a general-purpose personalization

engine. Instead, it is specifically for scenarios that involve a sales process in general and is specifically targeted at B2B sites that have a large number of products and variations. Also, it is not really straightforward, at least not yet, to customize personalization parameters or to configure new rules-based content targeting.

Former Clickability was one of the first to offer built-in functions so that site visitors could author user-generated content. To do this, you set up specific content types, such as “alumni news,” or “calendar events,” and the product establishes a “submission queue.” This content can either be relegated to this submission queue for review or it can go to the editor’s default inbox via standard workflows. As an alternative, you can set user-generated content to go live automatically on the website, without internal review. The company has enhanced the UGC feature by adding a spambot checker, to help filter out potentially inappropriate or malicious submissions through the external site’s submission forms.

Note that in either case, by controlling the content delivery environment, Clickability can conveniently account for user-generated content more effectively than a fully decoupled system, where the authoritative management repository resides in a different environment. Also, you can apply commenting to any content type, and not just “social” types like blogs, which is very nice.

Clickability also offers other useful site management applications, like built-in polling, subscription-based newsletters, and ad hoc email broadcasts. There’s an ad management tool, and while customers like the existence of this feature, some feel it’s not as useful as it could be, since reports can’t be tailored to a particular advertiser. There’s also the ability to detect requests from mobile user-agents, so you can conditionally change your output. For mobile, the company introduced a functionality called the “User Agent Redirect Component.” Its purpose is to provide mobile browser recognition so that specific rules can be applied to incoming requests.

The newsletters are worth special mention, since users can “subscribe” to particular categories of content, and site owners can create special mailings from the repository and put them through a workflow. Indeed, Clickability has several major customers who use the product solely for newsletters, taking advantage of the large tag library to customize their mailings for different categories of subscribers, while testing and analyzing different offers.

Built-in site analytics are a big part of the product’s value proposition, with in-context reporting residing directly within the content manager’s console. This creates an interesting opportunity for sites where the marketing team is closely aligned to content publishing priorities. Users with appropriate permissions can run traffic reports from within the CMS and act on the results right there. Using the built-in analytics, users can analyze traffic data based on content life cycle. The data includes information about advertising performance, campaigns, click-throughs, mailings, newsletters and so forth.

Customers of late seem less impressed with this service in action. Clickability doesn’t keep up with all of the features of packaged traffic-analytics packages (such as click-path analysis). On the plus side, some of the social media tracking and search log analysis reports are decidedly useful.

In addition to its CMS offering, the company offers a social media package with such features as user commenting, ratings, and user-generated video submission. In concert with the rest of the product, the social media package includes integrated analytics options. The collected

social media analytics can build custom profiles that get Clickability one step closer to offering true personalization capabilities. Current Clickability customers (who are not yet utilizing social tools) see the package as an opportunity for them to get started. More experienced customers, however, have found it cheaper and easier to go with free or inexpensive, best-of-breed tools to handle their comments, blogs, and video.

You won't see much beyond some social publishing capabilities (i.e., Facebook and Twitter), and light syncing with Salesforce.

Ancillary Services

Clickability uses a modified version of Apache Lucene for site search. It's nicely integrated, but perhaps not as good as your existing search engine. Test carefully. The Lucene-based search engine is heavily customized; it can search by content types, workflow status, and date ranges — just test to make sure it fits your needs.

Vendor Intangibles

Like CrownPeak, Clickability is known for its good support. Customers report extremely fast turnaround on inquiries, and a highly organized system of resolution and billing that includes hourly service fees that are billed in scant, five-minute increments. “Even when they make a mistake,” notes a challenging customer, “they work hard to set things right.”

After some complaints of longer than normal response times and a rare service outage, the company hired a new customer service manager who revamped their support procedures and several customers expressed that service levels went “from good to excellent.” In response to enterprise customers’ demand for more than just 24x7x365 monitoring of the system, Clickability established a dedicated phone number — accessible globally — where customers can call to receive ETA on resolution times.

One possible shortcoming (according to at least one customer), it is that Clickability’s support team cannot always maintain and troubleshoot the more advanced customizations that were performed by their own development group. This becomes more of an issue since Clickability relies more on their professional services for customer implementations. Customers say their professional services team knows their products well and according to one customer, they are honest in their explanation of what can be implemented and what cannot.

The company has pricing tiers that segment potential customers into Express, Professional, or Enterprise levels. Some of the base features of each tier are as follows:

Express: 5 seats, 1 domain, 1GB storage/month, 250,000 page views/month

Professional: 10 seats, 5 domains, 2GB storage/month, 500,000 page views/month, plus Social Media modules

Enterprise: 15 seats, unlimited domains, 5GB storage/month, 1,000,000 page views/month; also includes API and SDK, as well as Social Media modules

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

This pricing may be difficult to compare to other WCXM vendors, since most competitors don't include hosting in the mix. Express tier customers should expect monthly costs to start at \$3,000/month; Professional tier at \$5,000; and Enterprise tier at \$8,000/month. Of course, these can rise substantially. For example, an annual subscription at the Enterprise tier will likely cost you more like \$200,000+. If you don't need many user seats, however, the price can go lower. This level of pricing usually includes all bells and whistles: content management environment, content delivery environment, visitor engine, and other enterprisey features.

As usual, implementation costs are extra. For a larger Enterprise implementation with certain complexities, expect to shell out upwards of \$300K for professional services (starting at \$15K for a bare-bones implementation). The famed Content Connector customizations run at approximately \$60K for one instance. At least one customer found that the majority of the professional services hours were spent on working with the highly complex APIs, though to be fair, other customers have employed the vendor as a quasi-integrator to avoid using internal IT resources.

As your web presence matures and website(s) grow, be careful of rising costs. One customer we talked to had quickly expanded over their initial licensing and faced continuous renegotiation and billing issues.

Conclusion

Compared to its competition, Upland does offer a more rounded set of features for areas in digital marketing and experience management — be it analytics, ad serving, marketing campaigns, and various other bells and whistles. If you have scenarios in Basic Digital Marketing sites, or Landing Pages and Microsites — and you need web content publishing all from one vendor — then consider Clickability. However, if you prefer best-of-breed modules for these services and want to host your own website, Clickability would not be a good match.

By the time the new owner Upland figures out what to do with Clickability, even customers interested in exploring this SaaS CMS are likely to remove it from their shortlists. In general, it will take Upland a lot of effort to reinstate Clickability's former position in the market. Proceed with caution.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Magnolia: Magnolia

magnolia-cms.com

Vendor at a Glance

Specsheet	Magnolia: Magnolia 5.3 Summary
Geography	Mostly EMEA, some North America and Asia-Pac
What's New	<ul style="list-style-type: none"> • Single interface for touchscreen and desktop editors • Personalization/segmentation module • API for integrating external media assets
Strengths	<ul style="list-style-type: none"> • Decent editorial features and comparatively friendly contributor UI • Comparatively open architecture and support for multiple templating engines makes the product easier to extend • Localized in an impressive 16 languages • Strong repository services helps support very large sites • Company focused on standards adherence • System and company are comparatively developer friendly, and tech support is unusually well regarded • Optional "Enterprise" edition brings more functionality for a fee
Weaknesses	<ul style="list-style-type: none"> • Platform is very page oriented; not well suited to content reuse or even simple, metadata-based index pages • Workflow is rather clunky and requires developer to model • Surprising lack of services to support global publishing and distribution • Thin asset services mitigates against media-heavy sites • Very few built-in capabilities to support digital marketers, including a dearth of connectors to third-party marketing platforms • Thorough understanding of Java and product-specific expertise required; it is easy to make poor technical decisions • Community is sparse in some areas of the globe; It can be very difficult to find experienced specialists in North America
Potential Fit	Informational, Ultra-Large Single Sites
Unlikely Fit	Global/Enterprise Digital Workplace, Advanced Marketing Portal
Compare To	eZ Systems, Hippo, Adobe, e-Spirit
Operating Systems	Linux, Mac OS, UNIX, Windows
Repository	JCR (Apache Jackrabbit)
Client	Any browser
App Platform	Java
Licensing	Dual: GPL, or commercial license starting at US\$12,000 per year per server
Ownership	Privately held; 65 employees

Summary

Magnolia is a Swiss-based Content Management System offered under a dual license (commercial and open source). As with most open source, Java-based systems, the community remains comparatively small — but highly passionate.

Magnolia differentiates itself mainly through its support of Java standards, in particular the Apache “Jackrabbit” Java Content Repository (JCR). In fact, the platform is something of an anomaly: very developer centric, but not very complex or functionally rich. Your developers will like it better than your editors will — and much better than your marketers.

Magnolia has seen limited experience within the enterprise (e.g., for intranets), where it offers very little out of the box. Consider Magnolia for simple, mid-range, or departmental scenarios where the product is battle-tested, and its

bundled reference implementation offers a substantial head start. Large sites with mostly static content and fewer needs for interactivity (e.g., public sector, higher education, and print media) can be successful with Magnolia if they invest thoroughly in learning the platform.

For complex scenarios, the product may be extensible, but you should expect longer project cycles and the platform easily could come up short on functionality. Note that outside Switzerland and Germany, you may struggle to find experienced Magnolia consultants, despite recent efforts on behalf of the company to expand globally.

Scenario Fits	
Simpler Site	
Informational	●
Microsites & Landing	◐
Mid-Range	
Basic Digital Marketing	◑
Mobile Site	◑
Community Oriented	○
Complex Site	
Advanced Marketing Portal	○
Global Enterprise	○
Multichannel Publishing	◑
Ultra-Large Single	◑
Digital Workplace	
SMB/Departmental	◑
Global/Enterprise	○

Introduction

Magnolia was released in 2003 by a company called Obinary, which changed its name to Magnolia International in 2006. The 65-person commercial company is based in Basel, Switzerland, with a US subsidiary and offices in Spain and China.

When the Java Content Repository (JSR 170) standard came out, Magnolia moved quickly to support the reference implementation using Apache Jackrabbit. With the early buzz and relative success of the JSR standard, Magnolia appeared on many buyers’ radar screens. However, outside Europe, the community remains thin, and forays into North America have been rather slow and halting.

Initially, Magnolia’s community grew considerably around brochureware implementations, but in late 2006, Magnolia (the company) introduced a new Enterprise Edition aimed at larger sites and more complex scenarios. With the launch of the Enterprise Edition, Magnolia also introduced a license fee — even though a free (feature-limited) Community Edition is still available.

You will almost certainly want to use the Enterprise Edition, because the community is not big enough to support major implementations, and Magnolia does not support the Community Edition. Ninety percent of the company’s revenues now come from support and licensing.

The latest 5.3 release came in late June 2014. Breaking a bit from tradition, the company started to focus on digital marketing in general, and personalization in particular.

Technology

Technical Administration & Security

Magnolia uses a comparatively clean J2EE architecture, running inside a web container (e.g., Tomcat or Jetty). Magnolia stores content in an Apache Jackrabbit Java Content Repository. The project moved quickly to embrace the standard when it came out in 2005.

Architecturally, this means that Magnolia, Adobe Experience Manager and Hippo CMS now resemble each other. Where Magnolia now offers a complete “Standard Templating Toolkit” (STK, as a quick-start for you to base your own site on), Hippo provides a “Hippo Site Toolkit” as a presentation layer. Both products often refer administrators to views of the underlying repository to change the parameters stored there directly. Meanwhile, Adobe has moved on to a newer and somewhat higher performing version of the JCR, codenamed “Oak.”

Out of the box, Magnolia comes bundled with Tomcat. However, it can be installed on any J2EE-compliant application server, with WebSphere or WebLogic WAR files available in the Enterprise Edition. Note the “J2EE” part: Magnolia uses Java Beans, and not just servlets and plain Java objects.

Magnolia is a “frying” system; your pages are delivered dynamically. Unfortunately, Magnolia does not yet support publishing to static pages (except for its caching, described later). Consequently, make sure to allocate time to sufficient performance testing for your project. Note however, that the editorial model remains very page based, rather than component based (see “Contributing Content” on page 641).

When you install Magnolia, it creates two instances of the application. In Magnolia lingo, they are called an “Author” instance” and a “Public” instance. As the names suggest, the Author instance is the equivalent of the management environment used by administrators and authors to create content and manage the site content. The Public instance is an equivalent of the delivery environments, and is used by visitors to see the website. It is possible, however, to use multiple machines, but you will want to configure the Public instance a bit differently to do so.

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

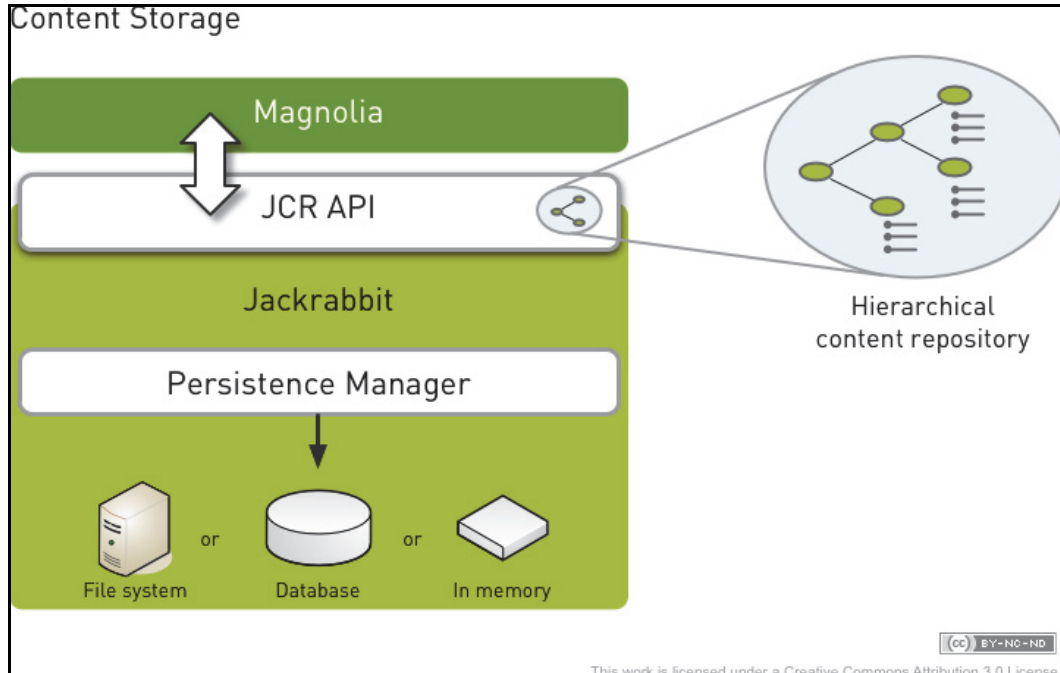


Figure 331. A simplified view of the Magnolia CMS repository. Remember that with JCR, you have choices for your persistence layer, although your choice will have performance implications.

Content gets published from one instance to another instance (or multiple instances), using a process called “Activation.” You configure one or more receiving servers, called Subscribers, where content is published. You can also schedule page publishing for a given date and time. Scheduled deactivation is also possible. The Enterprise Edition also includes what Magnolia calls, “Transactional Activation,” that is, if content fails to be published to one or more of the receiving servers, the update is rolled back, so you can troubleshoot the problem before you try again.

Editors can elect upon activation also to publish all child pages. This is handy for publishing entire sections or campaigns at one time. However, it happens asynchronously via messaging, since the processing could become arbitrarily long. You’ll want to set expectations with authors and marketers carefully.

You can install Magnolia in Amazon EC2, and some licensees have done so. Magnolia itself plans to offer an on-demand, managed service in the future.

User Management & System Reporting

To control access, Magnolia uses the familiar concept of ACLs (Access Control Lists), along with users and groups. Generally, you define a group and assign ACLs to these groups. You then define your users and assign them to these groups. The Enterprise edition of Magnolia can also integrate with LDAP and Active Directory for user management.

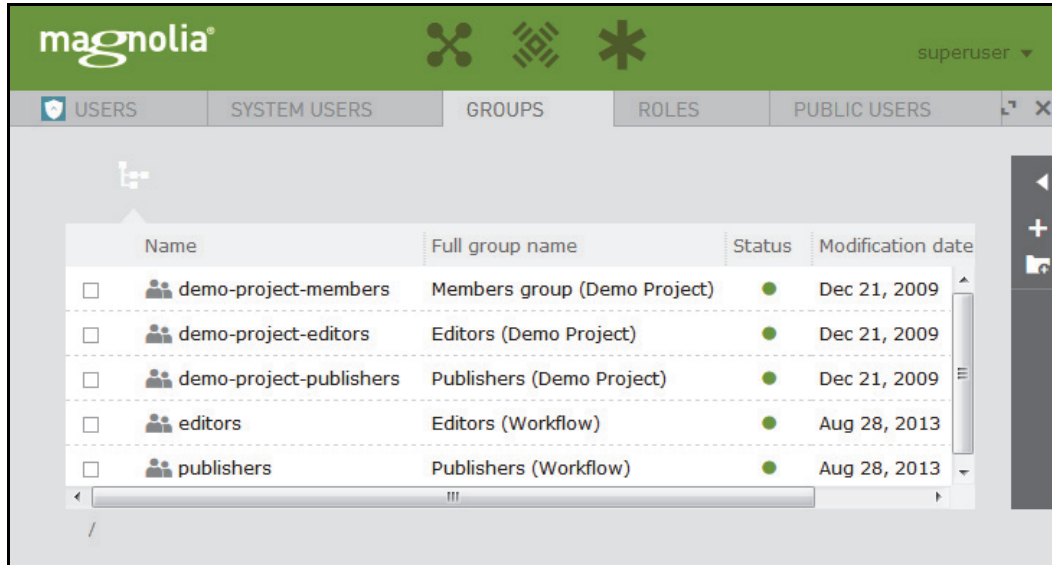


Figure 333. Group (above) and role management are powerful, but relatively simple.

This is one of the few areas where Magnolia has placed a nice GUI interface on top of the JCR. Customers laud both the power and granularity of role and group management, as well as its simplicity — a rare combination.

If you license the Enterprise Edition, you can use LDAP or Active Directory for authentication as Magnolia uses Java Authentication and Authorization Service (JAAS). JAAS is a set of APIs that enables services to authenticate and enforce access controls upon users.

Magnolia offers little in the way of system reporting. There is a list of “accesses” in the administration UI, which shows when users logged on to the system. The UI also has a “log viewer,” which shows more detailed information on various activities, but these only show the unformatted log files as plain text. Customers have complained about parsimonious logging; overall, this isn’t a report-friendly platform.

Development

Developers can use their IDE of choice to work with the templates, which are written in JSP, Freemarker, or the Java-derived, object-oriented scripting language, Groovy. This is a rather developer-centric platform. For example, you need a developer to create new content types (something nearly every other system enables in a browser). As one customer said, “Magnolia just assumes you have a high level of Java knowledge.”

Developers use a separate engineer-build environment to push code separately from content and browser-based configuration. Unlike Adobe, there are no content-plus-functionality “packages” that you can move through a combined DTAP process.

As previously discussed, Magnolia generates pages dynamically at runtime via templates. A Template defines the structure of the page and consists of multiple “Areas,” which in turn define an area on the web page. Think of Areas as template components that get rendered as separate <div> elements on your page (this supports some drag-and-drop functionality in the page-builder interface, which we’ll review later). To that extent, you can nest them, but unlike

higher-end systems, Magnolia doesn't make it particularly easy to track what Templates incorporate which Areas (nor vice-versa), so ongoing management can become a hassle.

The "Standard Templating Toolkit" comes with the Enterprise Edition, but can also be downloaded separately for the Community Edition. It's a reference site as well as a framework for Magnolia customization and development that tries to embed some best practices.

You can change the reference sites' look-and-feel via CSS, and styles are managed items in the system — but you should be aware that stylesheet modifications would not be trivial.

Likewise, while the template components easily can be edited from the interface, you must either really like what Magnolia has to offer out of the box, or in all likelihood, dig into the template code. Thus, most customers base their implementation off the reference STK; keep this in mind if you plan to come to the party with some very original (read: non-blocky) design. In fact, Magnolia has a tendency to refer customer problems back to solutions in the STK, which is inconvenient for those who had good reason to stray outside its confines. Fortunately, the latest STK natively employs responsive design.

The system isn't well suited for complex multisite setups, but it's possible to manage different sites (and their domain names) from the administrative UI. The sites usually are subsets of existing content (e.g., a specific language), and can have their own CSS. The functionality should be comfortable for basic scenarios, but don't expect the same sophisticated performance of platform systems.

Finally, note that (with some recent exceptions), Magnolia does not boast as many out-of-the-box connectors to other systems like CMR and email platforms. Customers seem to code these themselves.

Version 5.0 came with a revised "app framework," which looks promising. From within the admin control panel, you can search and download available apps from app.magnolia.com. To date, these seem to be mostly simple code snippets. Magnolia itself does not validate the contributions, so like any third-party module catalog, you'll want to vet licensing, performance, security, and such carefully.

Performance

A few licensees have reported performance problems in the editorial interface, in particular with 10+ simultaneous contributors. Scaling the editorial environment is an area where experience is limited, so you'll need to allow additional time for testing. Magnolia says it has improved performance with each dot release. We suggest you test this carefully, as you would with any system.

On the delivery side, Magnolia has its own page caching mechanism, which is configurable, but otherwise simplistic. For example, if you update an image, you cannot flush the cache on dependent pages. The company says you can plug in your own homegrown caching mechanism, which we would not encourage you to do casually. The Enterprise Edition comes with predefined caching scenarios.

We'll note here, though, that Magnolia receives good marks by licensees when it comes to scaling in the delivery environment.

Content

Contributor Experience

The initial start-up screen displays a Windows Metro-style set of decks in a touch-friendly motif. It includes useful notifications and very handy shortcuts to common tasks. However, once you select your path, you end up in a fairly traditional — maybe even a bit dated — WCM user experience.

Like most WCM tools, you’ll find two editorial environments: a tree-based back-end called AdminCentral, and an in-context editorial and page-builder interface.

In the tree view, icons on the right enable you to act on (e.g., check out, move, edit, delete, etc.) individual items. It’s simple and it works, as long as you know where you’re going. Fortunately, the repository search works well.

However, customers have complained that they constantly have to return to this interface to navigate around or even to complete different tasks on the same content item. In short, the interface is a bit click-heavy.

Content	
Contributor Experience	
Overall Usability	●
UI Accessibility	●
Contributing Content	
Authoring & Transformation	●
Tagging & Taxonomy	●
Content Reuse	●
Media & Document Management	●
Repository Services	●
Content Lifecycle	
Workflow	●
Globalization	●
Archiving & Compliance	●

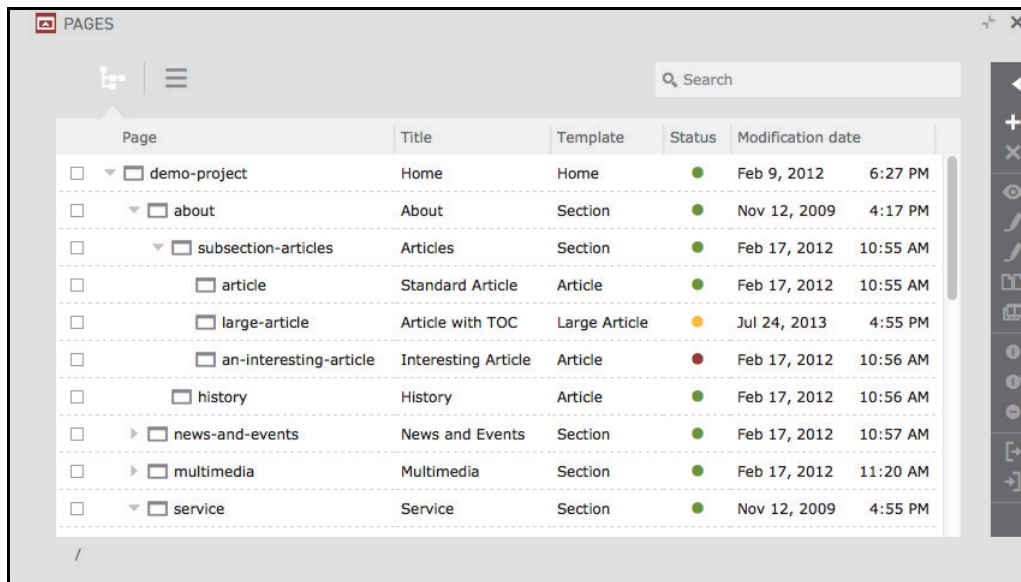


Figure 334. Figure 345. AdminCentral is a tree-based control panel with action icons on the right.

The AdminCentral interface is internationalized and now localized in an impressive 16 languages, including Russian, Chinese, French, German, and Spanish.

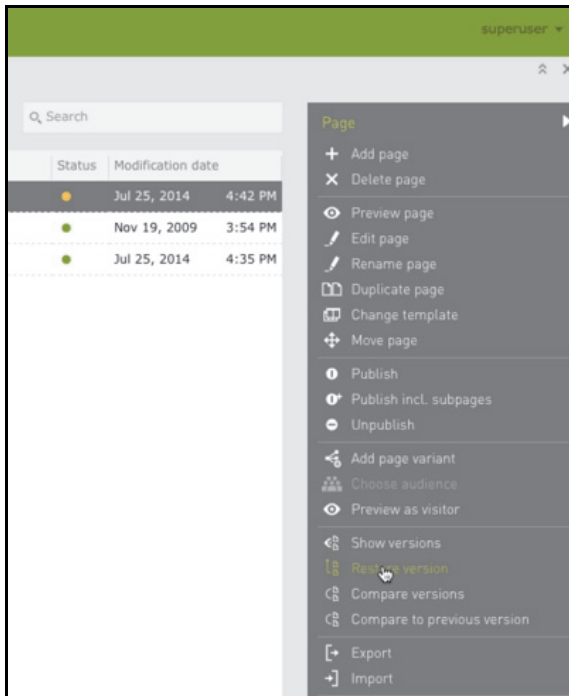


Figure 336. In the main editorial dashboard, you select a content item and then take an action on it. Pro: there's no annoying pop-up "sidekick" like Adobe and Ektron. Con: you need a fairly high-resolution monitor.

The page-builder interface allows you to work directly on the page itself, including its various editable elements.

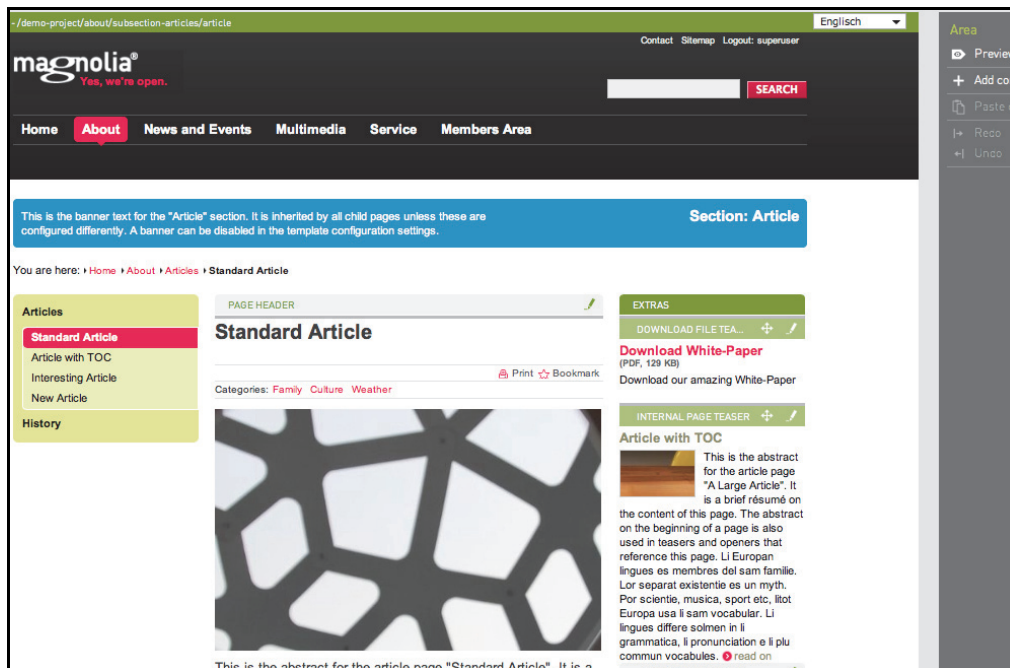


Figure 337. In-context editorial experience enables you to swap page components in and out readily.

Contributing Content

Your editorial colleagues will be working with content items that get rendered on pages in blocks where specified.

Editors open the page from AdminCentral, and get a preview with green edit bars. From there they can move elements around or start editing them (which will then pop-up a form).

A web page consists of multiple “Articles” (these are really just elements), and a content author uses a dialog to populate the content of an item by picking from a list of items in the repository, or entering the content directly.

You can add different components to any page via a simple picker. Magnolia offers a sizable number of standard elements available out of the box. These include elements for adding most popular types of content — text, images, links, video, list of documents, search, RSS, navigation, and many others. For rich text content like Drupal, you can swap in your own favorite editor.

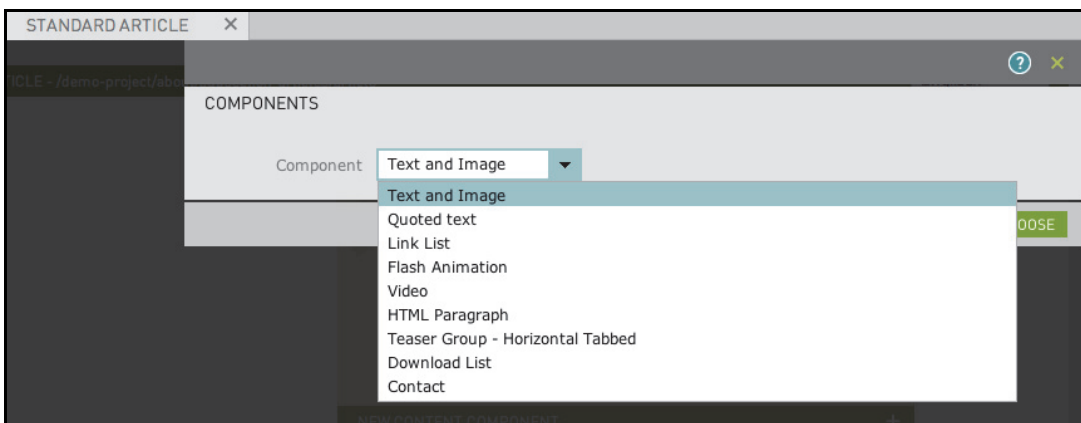


Figure 338. Adding (or replacing) a component in a template slot.

Versioning is specific to the Enterprise Edition, and enables you to review a simple visual version compare and roll back to previous page versions. However, the Community Edition stores only the previous version and it offers a simple “undo” function to roll back. In terms of version control, there’s a nice “soft-lock” mechanism that allows authors to break a lock when necessary with suitable notifications.

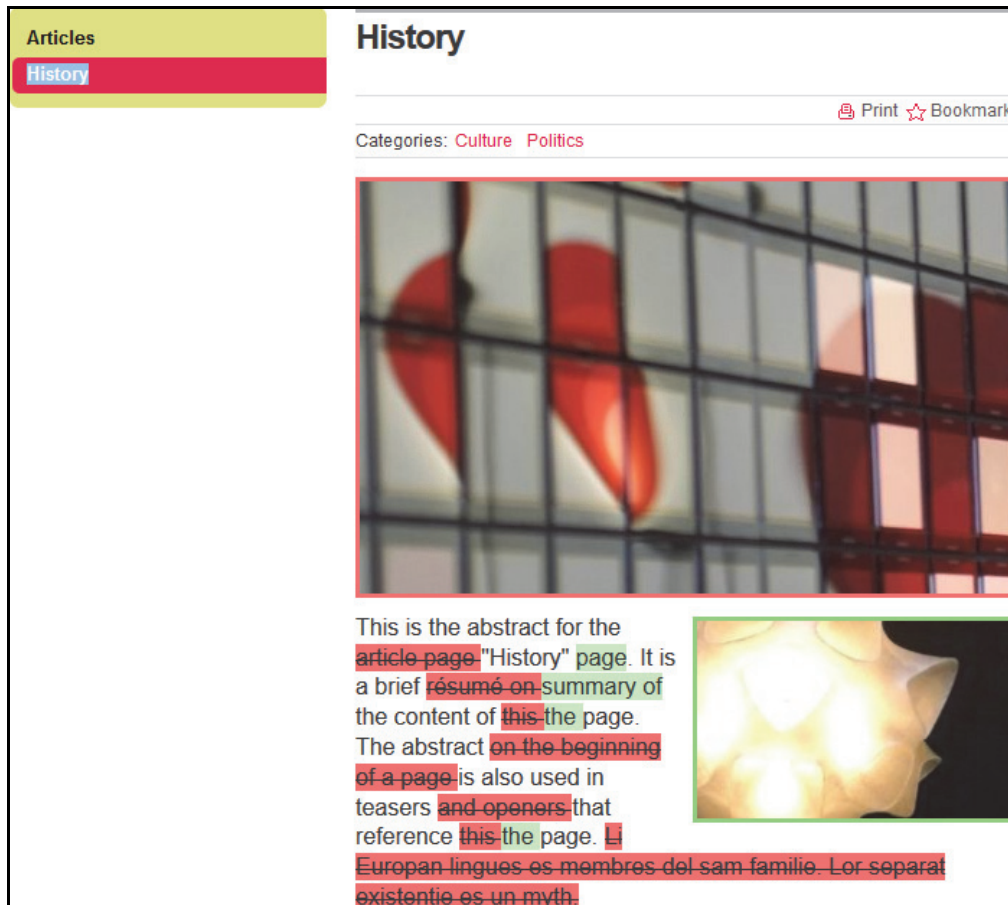


Figure 339. Version compare in Magnolia (Enterprise Edition).

Note that Magnolia is comparatively a very page-oriented system, which is not inherently a bad thing, since many web authors think that way and many websites (especially simpler sites) work that way. However, if you plan on intensive content component reuse, this system may not be ideal for you. Magnolia wants to push content to websites based on folders and sub-folders; it does not want to “pull” content at runtime according to variable parameters — at least not without a boatload of custom programming (which we don’t recommend).

This means that you can’t readily create automated, Drupal-like topic pages based on metadata, or other forms of “placeless” content. Creating lists of things tends to require editorial hand-crafting. With respect to taxonomies, the system oddly relies on flat lists of terms — with no tree structures — so metadata-driven experiences might be difficult to contrive in any case.

Magnolia (the company) has been claiming for some time that it would provide more reuse capabilities, but has been slow to deliver.

Asset management is a bit light, even after Magnolia added some connectors in version 5.3. Like most tools, you can edit images in your browser. However, the asset library services themselves feel somewhat thin and hidden. To create image collections (for things like slide-

shows), you have to move the images to folders manually. You can upload a zip file with multiple images, but there's no bulk tagging.

Content Lifecycle

Out of the box, Magnolia comes with a simple process that is triggered when a page is activated (published from author to public instance). True workflow (Enterprise Edition only) uses the OpenWFE engine, coming from an open source workflow project. It's a bit clunky: like Hippo, it favors stricter approval chains than typical web team collaboration. You have to back out of the workflow to see version differences (a big deal with translation workflows). You also need a developer to model the flows themselves.

For globalization, you can use the same content structure, and input content in multiple languages. This has been implemented by creating multiple nodes for each content node in the JCR. Magnolia makes this nicely accessible to editors, with a drop-down selection to switch languages in the interface. Languages also can be mapped to specific domains. Note that this still means that languages are mirrored (with a fallback language). For instance, if you create a French site based on the English site, it will automatically be the same as the English site until you manually add in French content. The two sites would be otherwise identical. This mechanism is limited, but practical for simple scenarios.

The challenge for you comes in managing localization processes within what is a very non-object-oriented system. With no notion of parent-child relationships, it becomes difficult to track inheritance and trigger changes in downstream (localized) versions of content based on edits to the parent. Unlike most other WCM tools, Magnolia doesn't ship with any pre-built localization workflows, though it's unclear exactly how they would work.

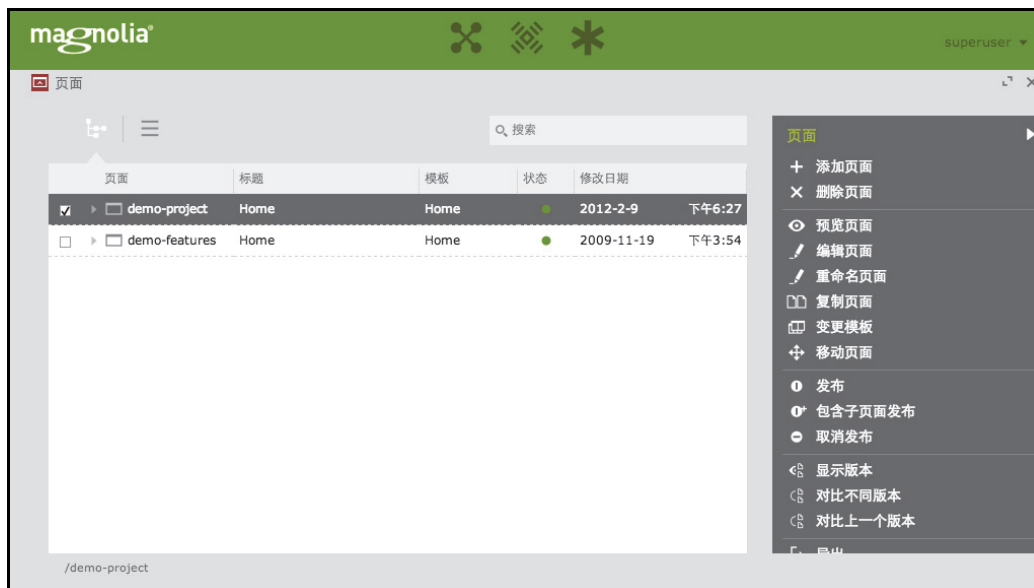


Figure 340. The Magnolia interface itself has been localized broadly, but the community does not seem to have put much effort into supporting localization processes for content.

To be sure, Magnolia can point you to large multinational organizations using the product, but we know that at least some of them find this aspect of the platform less than satisfactory.

Unless your publishing model is highly centralized or completely decentralized, you will find it a bit painful to manage the localization process.

Experience

Publishing

Even though Magnolia exclusively delivers dynamic content, it does so with a default .html extension, such as:

*www.magnolia.info/en/products/
 enterprise-edition.html*

For front-end development, the STK (templating kit) add-on has been designed with accessibility and SEO in mind. Therefore, the STK is not a bad place to start. If you can constrain yourself to CSS styling, you'll forgo some of the flexibility of HTML development, but you'll have a quick start in return.

Magnolia supports syndication in two ways. You can either syndicate content to another Magnolia instance using Magnolia's subscription mechanism, or by using RSS feeds. Each page created can be automatically exposed as an RSS feed.

Experience	
Publishing	
Standards Adherence	<input type="checkbox"/>
Multichannel	<input type="checkbox"/>
Mobile	<input type="checkbox"/>
Digital Marketing	
Site & Campaign Analytics	<input type="checkbox"/>
Testing & Optimization	<input type="checkbox"/>
Segmentation & Personalization	<input type="checkbox"/>
Social Media Integration	<input type="checkbox"/>
Promotional Campaigns	<input type="checkbox"/>
Community & UCG	<input type="checkbox"/>
Workplace	
Collaboration & Networking	<input type="checkbox"/>
Dashboard	<input type="checkbox"/>
Ancillary	
Site Search	<input type="checkbox"/>
Online Forms	<input type="checkbox"/>
Module Ecosystem	<input type="checkbox"/>

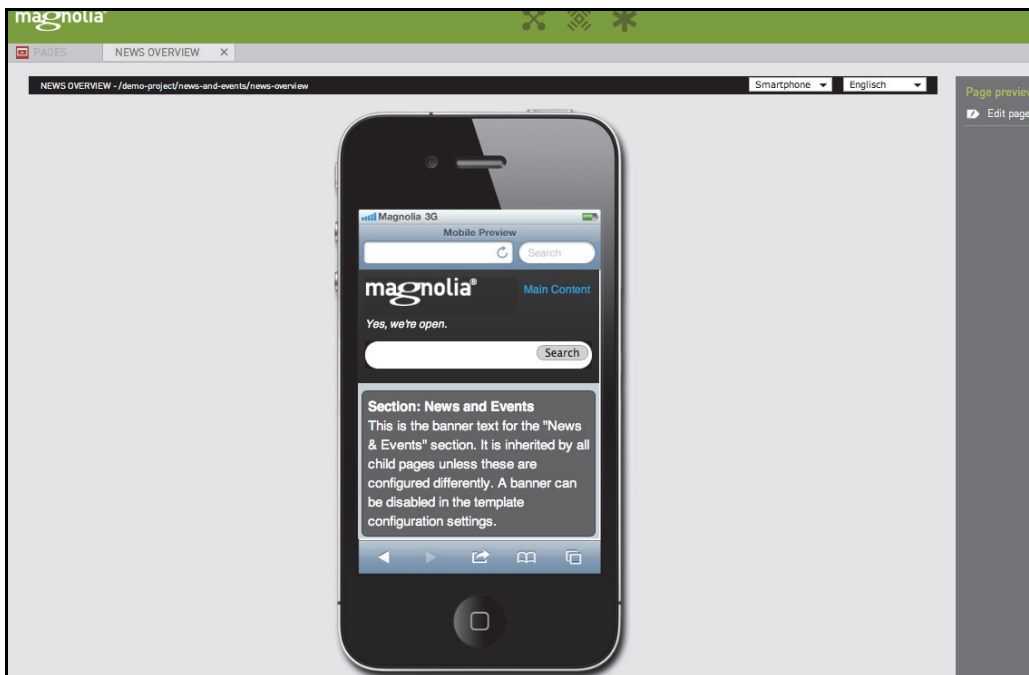


Figure 341. Magnolia includes the requisite mobile preview; as you might expect, the platform tends to favor responsive design over adaptive design.

The Enterprise version adds contact forms and user commenting. User comments can be approved through the AdminCentral UI; visitors can also register with the system, so their comments will be accepted automatically. Note that this functionality is quite simple compared to what most blog software offer; there are no spam/naughty filters. If you want to add a blog to your site, Magnolia might suffice, but if you anticipate heavy interaction with visitors, you should probably consider other options.

Digital Marketing

There is no native traffic reporting in Magnolia; rather, the platform enables you to integrate Google Analytics (both tags and reports) into the editorial interface.

New in Version 5.3 of the Enterprise Edition (and likely to be expanded in the future) is a profile-based personalization subsystem. It’s simplistic, but that’s probably a good thing — particularly since it is brand new and the community has had little experience with it. Note that you don’t personalize inherent component or template behaviors as you do in more advanced systems; instead, you define personas and then create article “variants” that swap in when that persona visits. Variants are complete article clones; if you modify the original article and you want those changes to cascade to all the variants, you need to copy it manually to each variant.

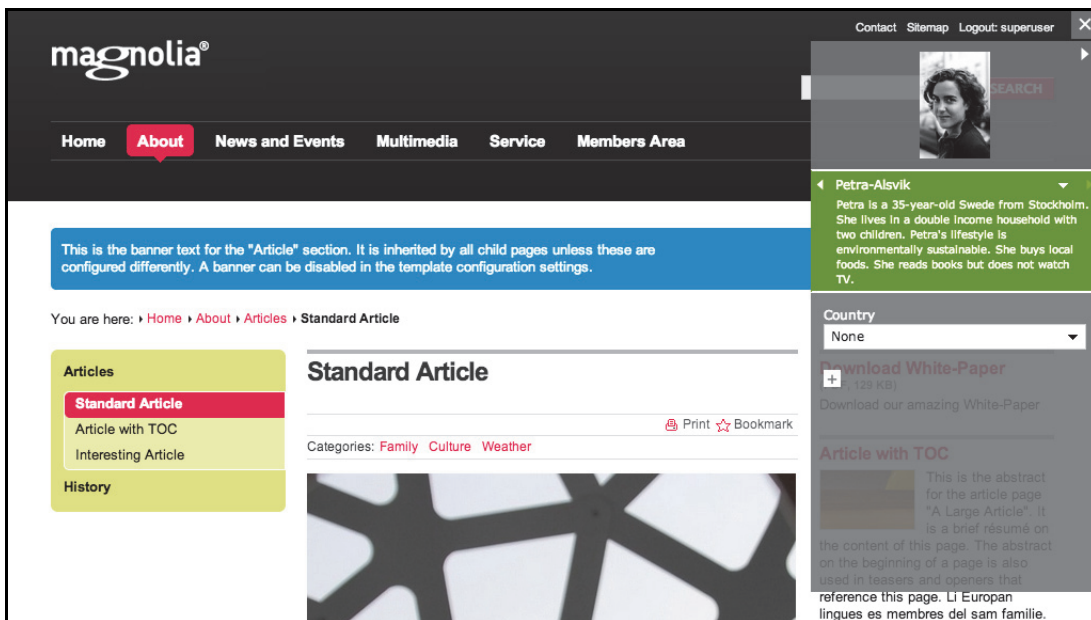


Figure 342. Assembling a page geared to a specific persona.

It is really more of a segmentation system, and the interface includes some nice impersonation features to preview different experiences in the editorial environment. Magnolia warns you that personalized pages don’t get cached, so consider performance here very, very carefully.

Ancillary Services

Search is available on the website tier via Apache Lucene, with optional Solr (takes a bit more work, but it’s documented). That’s good.

Magnolia supports a “Magnolia Store” for add-on modules, much like the currently popular app stores other systems are adding. However, since the community is quite small and there’s

only a limited number of partners, don't expect to see more than a few dozen additions available there.

In particular, you'll find a dearth of connectors to third-party marketing platforms.

Vendor Intangibles

As a JCR-based, open source Java system, Magnolia has some popularity among Java developers that gave the package a whirl.

However, the active community remains comparatively small — if highly passionate. The community seems very helpful, but it is still primarily centralized in few regions (Switzerland and Germany), with few experienced consultants outside the area.

Magnolia the company offers paid support and generally receives very good marks from customers in this regard — a rarity in this industry. Although growing, Magnolia gives off a family-run, small-business vibe, where employees are unusually accessible to customers. The company has support teams in Switzerland, China, and the U.S. Magnolia hosts a large and well-regarded user summit in Basel every year, but doesn't seem to work hard to build user groups elsewhere.

The company has grown a “certified partner” network, which they say now totals 60+ integrators, including several with unusually deep experience in the platform. Again, these tend to be more readily found in Europe.

As with most other open source projects, Magnolia documentation is a bit thin. The material that exists is relatively good, and you can readily find community wikis and forums as supplements. Just beware that the documentation tends to lag behind the current version.

Magnolia has accelerated its release cycle lately, with dot-releases seemingly every quarter, and bug-fix releases on an almost monthly basis. Some dot-releases are bigger than others are; exercise some caution here, as customers have reported some difficult upgrades. Once again, official support from Magnolia will help you substantially here, though we know some larger licensees dispensed with paid support after their first year.

Magnolia has three different licensing models. The vendor uses a typical commercial open source model here; no up-front license fees, but you pay for annual support:

- **Community Edition** – Open source (GPL), which is free but unsupported and limited in features and licensing. The edition does not offer LDAP support or many other niceties. You can add the STK, but are required to display a Magnolia logo on every page you publish if you do.
- **Enterprise Edition** – Starts at \$12,000 annually per server with unlimited support, it adds LDAP, advanced search, and JBoss support. The STK can be used without attribution.
- **Enterprise Edition Pro** – Starts at \$22,000 per server, it adds WebSphere and WebLogic deployment, as well as some advanced functionality, such as personalization

Pricing will vary based on your tier of license support.

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

The Enterprise Edition is licensed via the “Magnolia Visible Source” license, which allows you to modify the source code of the Enterprise Edition for your own use, but disallows distribution or deployment of the binary. You need to buy an Enterprise Edition license for every server on which you run your modified Enterprise Edition.

Note that the above prices are indicative; contracts are in Swiss Francs (CHF). “To keep prices low,” only EE Pro licenses are negotiable, though we suspect the vendor would be flexible in this for large EE deployments.

Conclusion

If Magnolia the company is solid and well respected, Magnolia the technology is a bit of an anomaly. It’s a developer-centric platform that nonetheless favors simpler editorial models and less marketing-intensive use cases. It can support very large sites, but excels at those where page- and folder-oriented publishing trumps visitor interactivity. Overall, this gives the platform a kind of amorphous, quirky quality; it does simple well, and can support very complex applications, but you’ll likely find it a sub-optimal choice for most in-between scenarios.

Magnolia is best suited for simpler scenarios and ultra-large single sites where styling the templates from the Standard Templating Kit makes it easy to deploy a new, mobile-friendly implementation quite quickly.

The lack of a bustling community makes the Magnolia ecosystem look bland compared to high-profile, open source projects like Drupal. Nevertheless, Magnolia the company is unusually well liked by its customers — particularly the developers it actively courts. Moreover, Magnolia has proven to be a highly stable system and is gradually improving. You’re not going to get sudden surprises from the Swiss engineers, and many customers seem to like it that way.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

CCI: Escenic

cci.europe.com

Product at a Glance

Specsheet	CCI: Escenic 5.4 Summary
Geography	Primarily Europe
What's New	<ul style="list-style-type: none"> • New video workflow in Content Studio • New JavaScript framework for building extensions to Content Studio
Strengths	<ul style="list-style-type: none"> • Broad support matrix • Strong fit for media companies, notably newspapers (where Escenic has developed substantial domain expertise) • Java-based architecture with extensive tag-library API • Content Studio thick client can be highly usable for power authors • Cross-media approach simplifies content reuse to multiple channels • Good support for basic approval workflows • Strong metatagging functionality • Many micro-application expansion modules available
Weaknesses	<ul style="list-style-type: none"> • For enterprises not in the media vertical, the learning curve will be steep and additional implementation support may be required • Feature set is more broad than deep • Workflow is a bit limited out of the box for enterprise use • Product is split into a profusion of separately priced add-ons • Linux servers are virtually a must in the delivery tier, unless you're not concerned about future scale-out • Complex and potentially expensive licensing scheme • Few consulting partners outside of the Nordics and Northern Europe • Scalability of most recent version may not be heavily tested
Potential Fit	Multichannel Publishing, Ultra-Large Single Site
Unlikely Fit	SMB/Departmental or Global/Enterprise Digital Workplace, Advanced Marketing Portal
Compare To	CoreMedia, Polopoly, Magnolia
Operating Systems	Ubuntu Server LTS, Red Hat Enterprise Linux (2.6), Debian GNU Linux, Fedora Core Linux, FreeBSD, Sun Solaris, Microsoft Windows 2003/2000 Server
Repository	Database: Oracle, MySQL, SQL Server
Client	Thick client (Java) that runs on Windows and Mac with limited, browser-based in-context editing UI
App Platform	Java/J2EE: Oracle, BEA, IBM, Resin, and Tomcat
Licensing	Complex, starting at €30,000 and going up steeply from there
Ownership	Privately held

Summary

A mature product with a relatively clean J2EE architecture, Escenic has slowly but surely built a loyal global customer base, mostly among major media customers, mostly in Europe. You'll find the product best suited for complex scenarios, particularly high-traffic Multichannel Publishing and Ultra-Large Single Sites, but potentially also Community-Oriented Sites. Mid-range and smaller initiatives will most likely find Escenic's features to be overkill — and overly expensive.

Escenic achieves scalability, security, connection pooling, operating-system independence, and other “enterprise” qualities by virtue of running in a J2EE application server environment. This also means you need in-house Java expertise, if you expect to get the most out of the product.

In late 2013, Vizrt sold off Escenic to a long-time partner and distributor — CCI Europe A/S. The three-stage transaction is estimated to total 4 million USD.

Scenario Fits	
Simpler Site	
Informational	<input type="radio"/>
Microsites & Landing	<input checked="" type="radio"/>
Mid-Range	
Basic Digital Marketing	<input checked="" type="radio"/>
Mobile Site	<input checked="" type="radio"/>
Community Oriented	<input checked="" type="radio"/>
Complex Site	
Advanced Marketing Portal	<input type="radio"/>
Global Enterprise	<input type="radio"/>
Multichannel Publishing	<input checked="" type="radio"/>
Ultra-Large Single	<input checked="" type="radio"/>
Digital Workplace	
SMB/Departmental	<input type="radio"/>
Global/Enterprise	<input type="radio"/>

Introduction

Based in Oslo, Norway, Escenic started with a clear media focus from its very beginning in 1997. Today, the company boasts 80+ employees, with development undertaken in Oslo and Dhaka (Bangladesh). The company also has sales representatives in New York, Hamburg, Bangkok, and London.

The 2008 acquisition of the company by Vizrt has finally started producing a noticeable impact on the business and products. Vizrt, which specializes in media asset management and broadcast graphics, counts well-known television channels as its customers (including CNN, CBS, Fox, BBC, Sky, and ZDF). Vizrt strengthened Escenic's focus on the media as one of its “storytelling tools,” alongside the existing Media Asset Management and broadcast graphics offerings. In late 2013, Vizrt sold off Escenic to a long-time partner and distributor — CCI Europe A/S. The three-stage transaction is estimated to total 4 million USD.

Some 600 media sites in 21 countries run on Escenic software and the company has acquired a great deal of publishing-industry expertise, including integrations with many of the commonly used newspaper production systems. Some of its largest and most complex implementations are with newspaper companies (including Aftonbladet, The Times Online, The Sun, The Daily Telegraph, Daily Mirror, The Independent, De Telegraaf, and NRC); although among their customers today you can also find a few government organizations. While still focused on the large Scandinavian home market, Escenic says it now receives roughly 60 percent of new business outside of that region, and recently has seen growth in South America in particular.

Escenic's R&D roadmap tends to be heavily swayed (not surprisingly) by the needs of media customers. In the past few years, this has led to separately licensed products and modules, such as the “Tip Engine,” “Event Engine,” and “Community Engine” (now merged into the core

product). These modules are examples of R&D efforts that, by focusing on the highly specific needs of certain kinds of customers, differentiate Escenic from competitors, but also make Escenic seem like less of a pure-CMS play. The Escenic platform will certainly support the development of many kinds of systems (in theory), but as a practical matter, the customers who are getting the best value for their money with Escenic seem to be media companies, membership organizations, and community sites.

Version 5.0 was released in late 2008, with an updated Content Studio (Escenic’s thick client), and it replaced search solution Lucene with Solr. Version 5.0.8, released in September 2009, added Arabic and Bengali language support, and functionality to find related content; from version 5.0.9 onwards, the 5.0.x branch is in maintenance mode (no new functionality will be added). Version 5.1 added the plumbing for the “Revision History Module,” which handles versioning of all content types. Version 5.3 added Escenic’s own bespoke HTML component framework for extending the (already quite busy) Content Studio interface with custom elements. 5.4 added workflow integration to Vizrt’s Media Asset Management tools with the additional support of meta-tagging engines.

However, you should note that many of the sites live on version 5 are new — not existing — customers. This means that it still hasn’t been heavily tested in the high-traffic scenarios that are typical for many customers. This should be due in no small part to the fact that tools for the migration from the 4.x branch were introduced after the 5.1 release.

Technology

Technical Administration and Security

Escenic is a Java product on the server side, with a standalone thick client that contributors and administrators can run locally on the desktop. The main component of the product is the Escenic Content Engine, which runs in a Java application server (Figure 343).

As with eZ Publish, OpenText, and some other systems, you can tweak a wide assortment of configuration settings through a single Java properties file for each website, as well as across websites.

Escenic uses dynamic delivery of pages, and handles both the management and multichannel delivery of content. In its home base of Norway, several Escenic installations are integrated with enterprise portals where Escenic is used to manage content, which is then delivered inside the portal using JSR 168 compatible portlets.

Contributors working on a website can be assigned different roles on different sections.

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

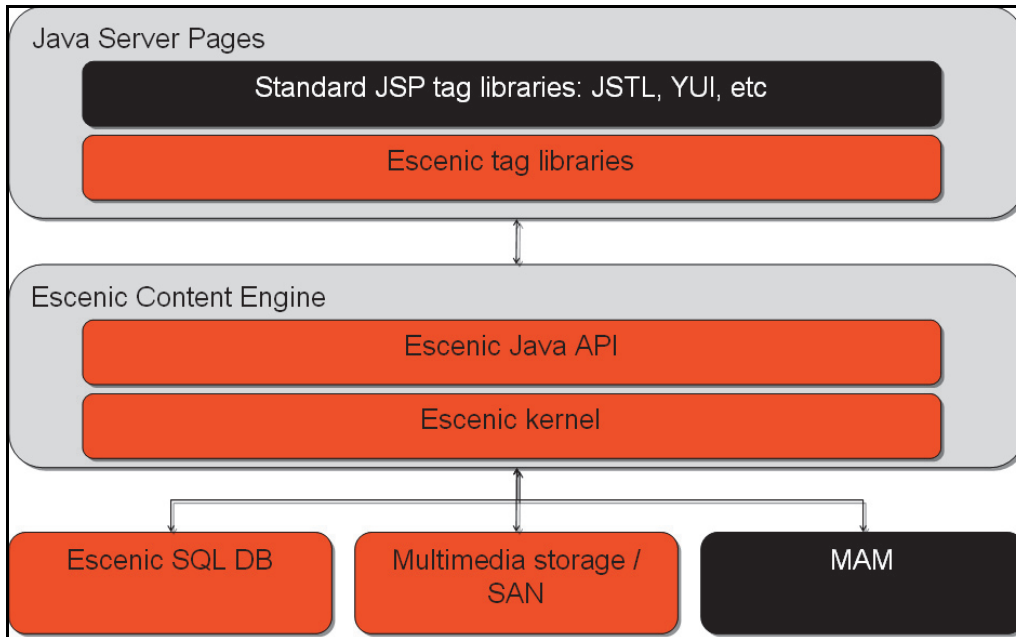


Figure 343. The Escenic Content Engine stores all content as XML in a database. XML content must conform to Escenic’s DTDs, which it will, if you create your content in Content Studio. Otherwise, you’ll need custom XSLT transformation steps to convert any existing or external XML content to an Escenic-compatible format. Source: Escenic.

Development

True to its media roots, Escenic calls each website a “publication.” Through syndication of content in XML files and presentation in XSLT, you can publish content to multiple platforms, such as mobile devices or other digital platforms.

Escenic stores all content as XML conforming to Escenic schemas (Figure 343). If you have preexisting XML content from another XML-based system — or you regularly ingest XML content from third parties, and you want it to be directly consumable by Escenic’s repository — you will need to make it compliant with the Escenic schemas. This could entail some XSLT development (perhaps a lot of it) if you’re moving XML content from another system to Escenic. There is an option of ingesting feeds from external sources (typically, RSS); this should make it easier to import dynamic content than in the previous versions, but you should test if this works for your scenario.

Escenic manages content in user-defined content types. As you could guess, a typical one is called *Article*. You define these content types in XML files.

Templates are JSP based and can be created and modified using your IDE of choice. As a developer, you can use Escenic’s tag library to ease the development efforts; Escenic says that it has cut the number of proprietary tags to half (and is moving to JSTL.)

You can potentially look into the Escenic’s Widget Framework to alleviate some of the development complexity. The framework allows the creation and placement of a range of standard components into pages, with some configuration controls available within Content

Studio. Depending on what content you wish to expose on the front-end, this has the potential to provide some development shortcuts.

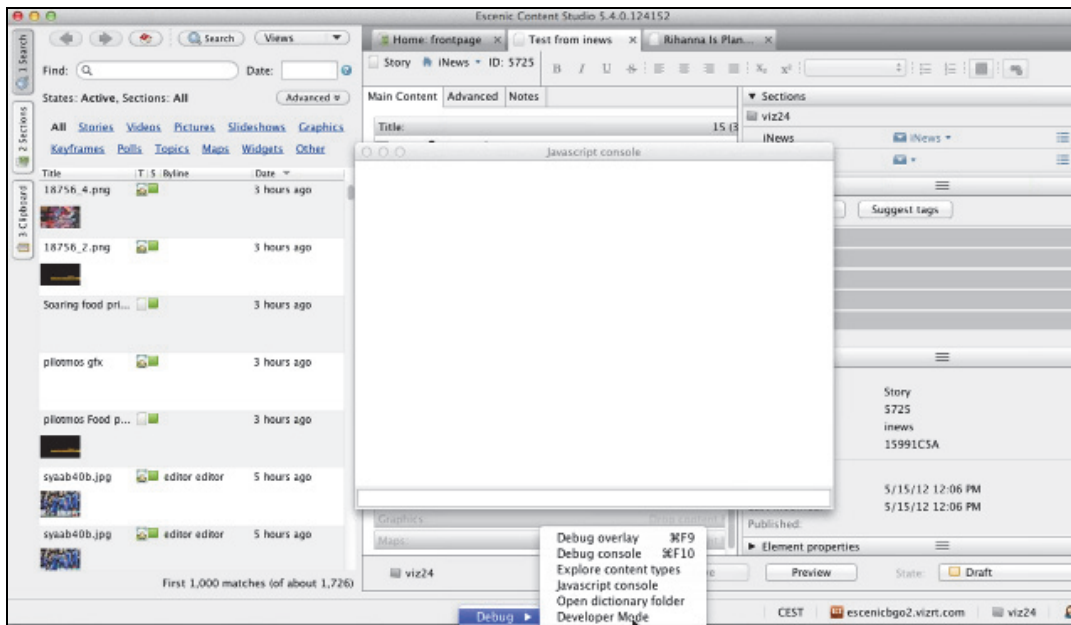


Figure 344. For developers, Content Studio can feel a little cramped.

Note that — as with content type definition — you need a Java developer with Escenic, even for small changes. Several licensees have reported that they are missing a more WYSIWYG approach to making templates for special events or topics. One customer summed it up by saying, “The system is doing what we need it to do for now, but because of our lack of technical knowledge, we’re not really able to utilize it to its maximum potential.”

To be fair, much the same can be said about many systems in this report, but you shouldn’t forget that while Escenic is quite purposefully targeting its media use cases, this doesn’t mean your front-end is a simple out-of-the-box affair; and customizing it isn’t a simple matter of changing the HTML. If you want to be agile in your presentation, you’ll need developers on staff.

Performance

As with most other dynamic delivery platforms, achieving high performance requires a bit of effort. Many of the issues involved in performance tuning and capacity planning are standard J2EE concerns. If you host a high-traffic site, budget for extra hardware and extra testing.

Escenic offers caching at four levels, which may help you achieve better performance:

- JSP fragment caching
- Presentation object caching
- Database object caching
- XHTML caching

The cache can be flushed, based on a fixed time or change event. In addition, Escenic has experience working with third-party external cache servers, such as Oracle WebCache or the

Squid Web Proxy Cache, and content delivery networks (such as Akamai, which Escenic says is used extensively by its customers). This is typically a requirement for their media customers (with important breaking news, a newspaper can't afford to have its website buckle under the peak load), and if this is relevant to your scenario, note that not too many vendors in this report can claim similar experience. (Drupal, for instance, is now oft-mentioned as a high-profile contender for interactive websites; however, the White House web team had to bring in Acquia itself to adapt Drupal to using a CDN, which hadn't been done before.)

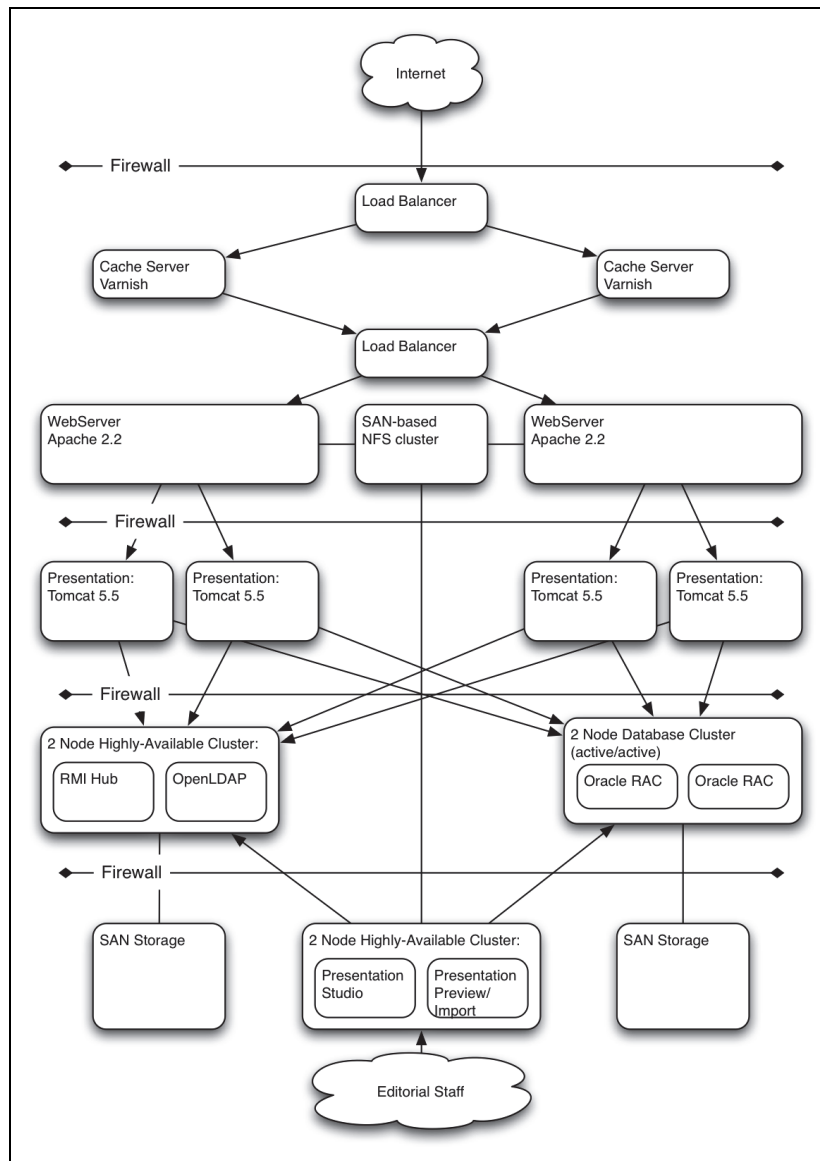


Figure 345. A sample Escenic architecture. Though the product offers plenty of alternative configurations, choosing the right one is complicated. Source: Escenic.

Escenic recommends use of the open-source Varnish platform. The latter is a sophisticated caching reverse-proxy engine, specifically designed for content-heavy dynamic websites (and,

in fact, originally developed for a Norwegian newspaper). Escenic used to be involved in Varnish development, and says it can deliver 10-20 times better performance than Squid.

The details of how Varnish works are beyond the scope of this review, but one thing you should note carefully, is that Varnish is aimed at FreeBSD 6/7 and Linux 2.6 platforms, in order to take full advantage of the virtual memory system and advanced I/O features offered by these operating systems. While Escenic proudly claims platform independence, this is one of the reasons you'll definitely want to use Linux servers for the delivery tier. Customers report difficulty getting the product to perform well on the Microsoft Windows platform.

Content

Contributor Experience

The preferred client is the Escenic Content Studio, a standalone, Java-based thick client, developed originally for Escenic's media customers. The client doesn't have to be deployed (it launches from the browser using Java WebStart), but note that the usual caveats for Java clients apply: If you have a lot of different client PCs, you could run into some tricky JRE conflicts, though we haven't heard of any issues specific to Escenic.

It's a busy interface that can be imposing to casual users, but could please power users who spend hours a day using it. By the look of it and the specific terminology used, the client does indeed have a distinct "newspaper" feeling. Version 5.0 has improved on clarity and usability (including tabs that allow you to have multiple items opened, as in recent versions of FirstSpirit), but the interface still works best on high resolutions. According to CCI, its approach to interface design is one of "gradual change," perhaps reflecting heavy reliance on newspaper clients, where production interfaces remain almost static for years at a time.

Escenic used to have a browser-based client, called Web Studio, but this is now a purely administrative interface. According to the company, users find the new Content Studio easier to use (and presumably, maintaining two separate but equally functional clients was a bit of a strain on the company's R&D). For browser access, editors can work with in-context editing (using "Simple Publishing"); though the vendor says there has been relatively little demand for this.

Starting in version 5.3, Escenic supported its own HTML component framework so that developers could customize the Content Studio interface with custom widgets. This could be useful for some integration tasks, allowing the ingest of content feeds from other systems, for example, which is a common requirement in newspaper clients. However, you will need to have strong JavaScript knowledge on your team, as well as the ability to master an additional semi-proprietary set of Escenic functions.

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

Content Studio is currently available in Danish, English, German, Swedish, Dutch, Arabic, Bengali, and Norwegian. For spell checking, Ispell dictionaries are supported, though the company warns that not all of the dictionaries have been tested.

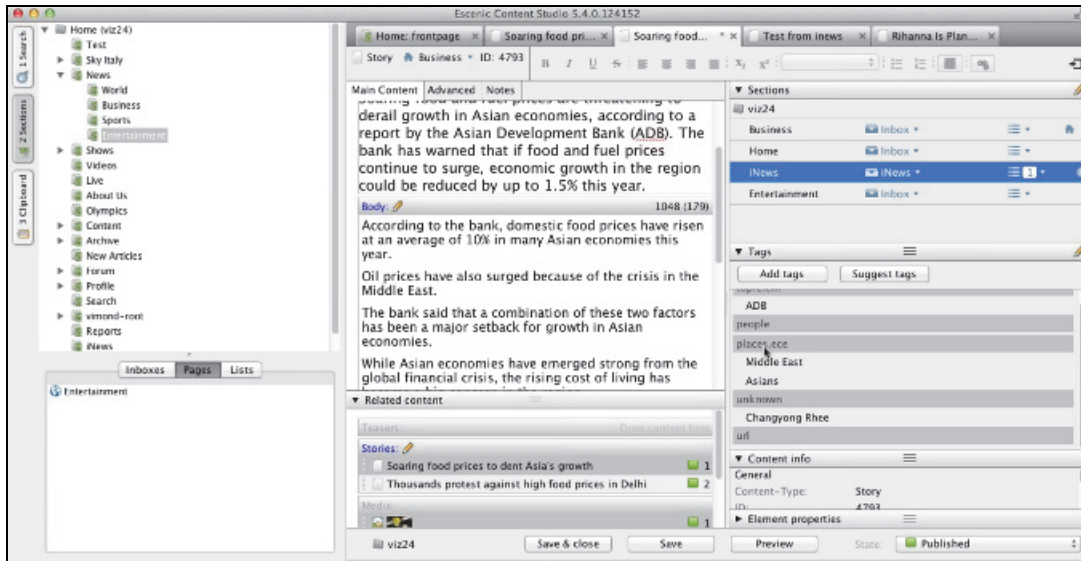


Figure 346. Escenic Content Studio's UI is capable, but busy.

Contributing Content

The desktop interface, Content Studio, has drag-and-drop functionality, programmable shortcut keys for all functionality, and image editing (Figure 349). Traditionally, vendors pushing their thick clients (such as FirstSpirit) have claimed it would be impossible to get the same features to run well in a browser window. With the maturing of frameworks for AJAX, it's becoming hard to maintain that, and several other vendors have now had a comparably "desktop-like" experience running inside the browser for years now (Sitecore, with its Vista-like desktop, is an extreme example). Still, there is some truth to Escenic's point that a thick client still has a slight edge. Note that it can be a hassle to get the client to install over slow connections (say, a decentralized group of authors, working from distant locations over dial-up), and the Java Webstart launch is of little comfort there. However, once it's up and running, it could consume less bandwidth than a complex web-based interface. Test and compare carefully if this is a concern for you.

Within the editing window, the text is structured according to the underlying XML definitions. Unlike most structured content editors, however, Content Studio doesn't emphasize this too much; it doesn't show the usual jumble of strictly separated form fields. The archetypical newspaper article, for instance, will have a "Title," "Lead-in," and "Body"; but on the screen, this will appear to be a coherent item to edit, using grey lines as delimiters. Related links, photos, and other references show up on the right; editors can search for them, then drag and drop them in the article.

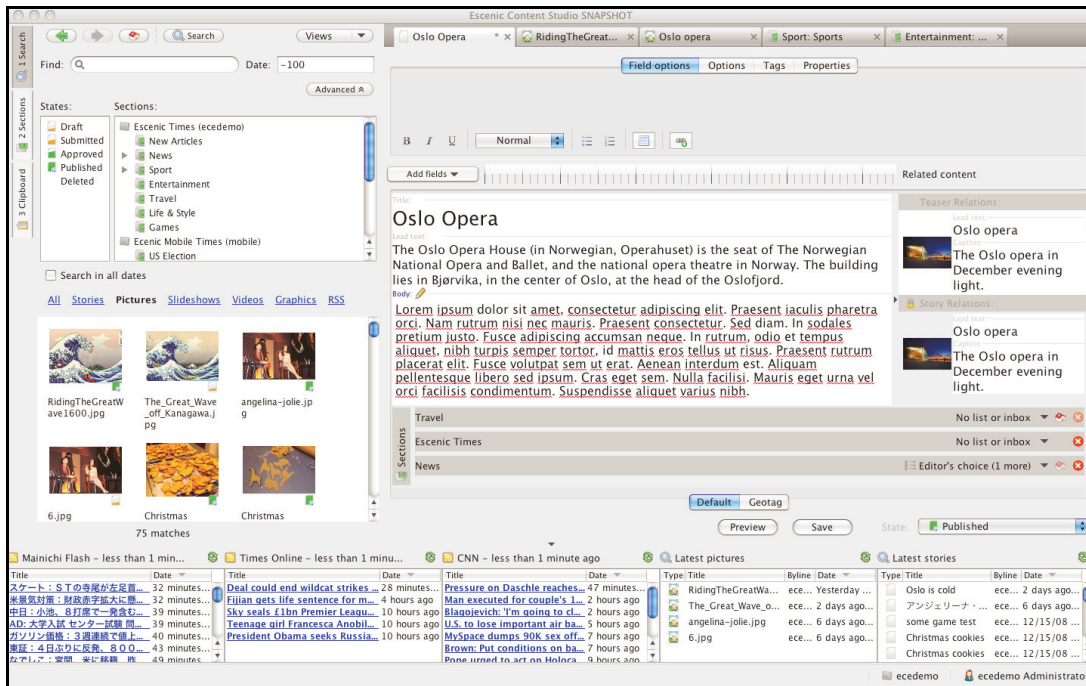


Figure 347. Escenic Content Studio was updated for version 5.0. Note the faceted search on the top left, and the live feeds on the bottom.

Editing pages is a matter of finding the right articles, and dropping them into the assigned slots (which are defined by the templates). This means that it's relatively easy to swap articles on a front page, add new ones, and switch them from a "left column" to a "right column." However, the process isn't exactly WYSIWYG, and supposes that editors already know what the actual page will look like. To see the effect of their actions here, they need to switch to the InPage view.

Escenic very strictly enforces the separation of content and design; that is to say, authors can't decide on rich text formatting, such as color or text size. Instead, authors must apply styles. Then the output channel (i.e., a website's CSS) will determine what the style will actually look like. Though in theory such a strict separation is one of the tenants of content management, it would have been nice to have the option. The InPage Editor allows you to edit front and section pages completely WYSIWYG, including color and positioning, and even some typographic details. However, this tool caters specifically to designers, who need full control over the layout and want to adapt the design daily — much as you would a newspaper front page.

There is no integration with Microsoft Word; again, the vendor maintains users find the Content Studio easier to work in. This may be true for editors in a company HQ, but for reporters on the move, the Java client doesn't support off-line editing.

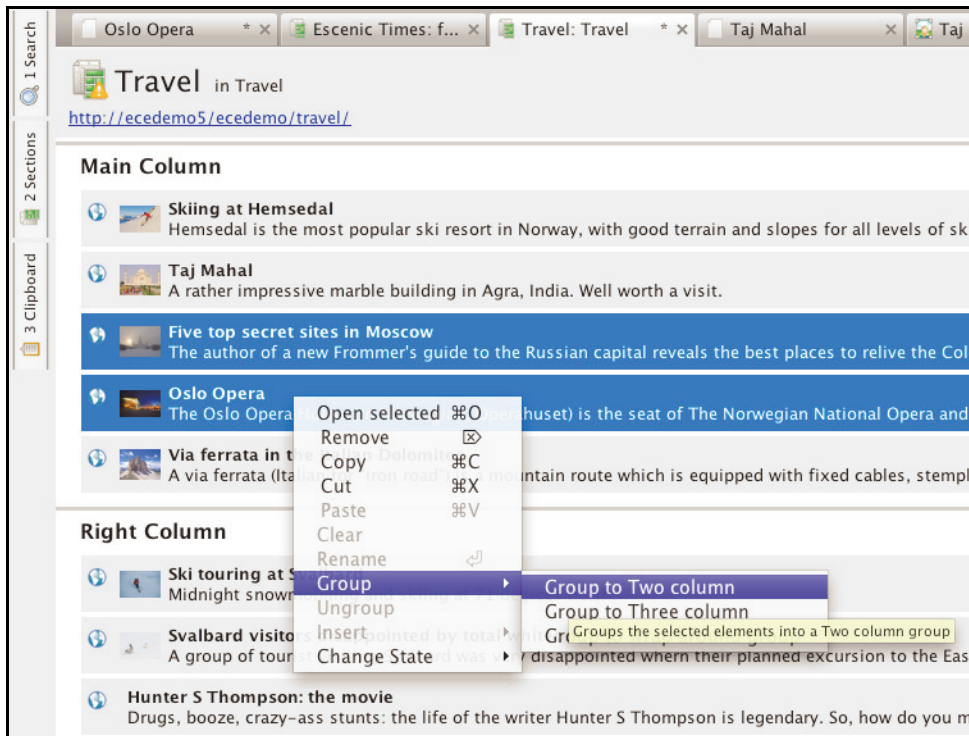


Figure 348. Non-technical staff can rearrange columns using Content Studio, based on the templates developed.

Content Studio includes basic image editing; you can crop an image and adjust brightness and contrast, but not color. For more complex actions, you can launch an external application (e.g., Photoshop) to edit the image. For many online publications this will suffice, but don't expect the system to perform as a lightweight DAM solution.

Version 5 switched from Lucene to (Lucene-based) Solr, and uses Solr's capabilities for faceted search. This means you can search on a variety of criteria, i.e., publish state, tags, title. Note that while the added functionality is comfortable to use, it's not as modern or advanced as other Solr implementations. On the upside, the search engine can also be used to suggest related content automatically.

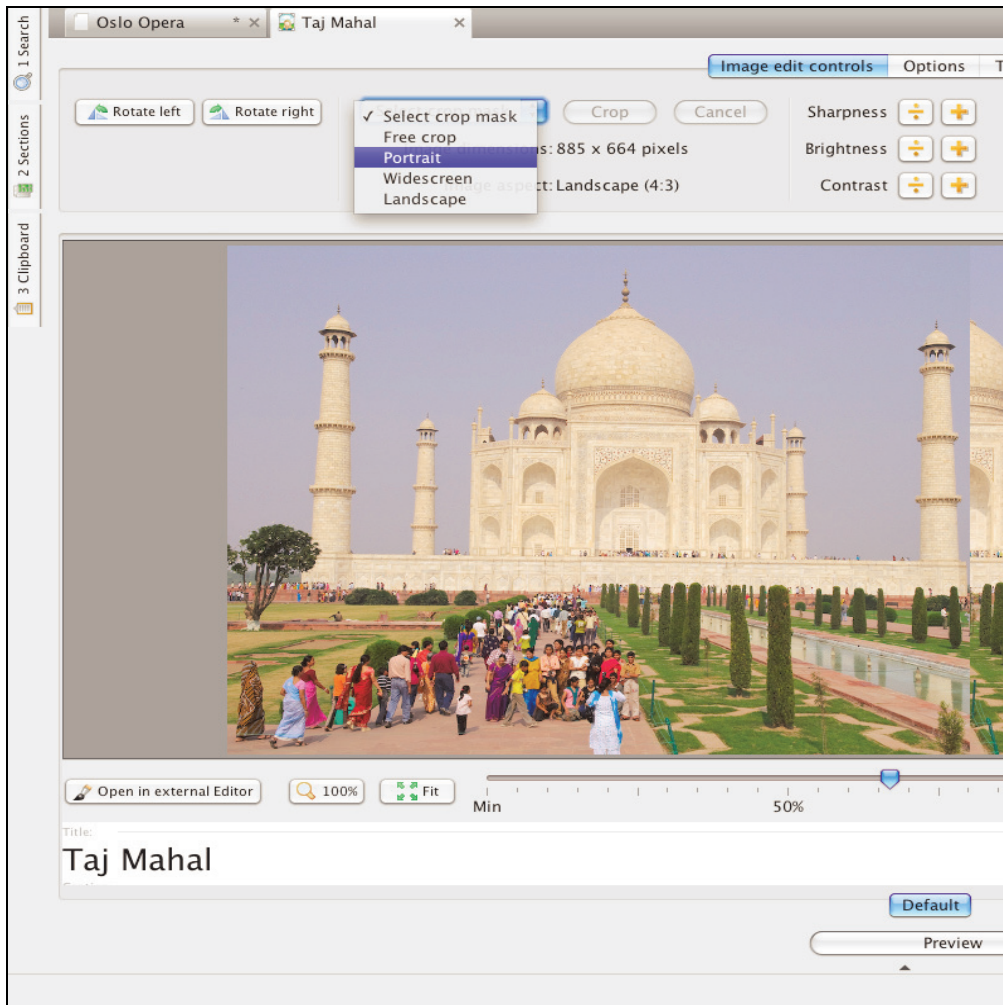


Figure 349. As an editor, you can perform basic image manipulation using Content Studio, or open the image in an external application.

Content Lifecycle

Workflow services seem to function well for Escenic’s core media constituency, but not so well for other types of enterprises. Version 5.4 adds a new, video-specific workflow, designed to be integrated with the Vizrt MAM platform. Approval processes within media companies typically favor speed over complexity. Escenic lives up to this need by offering a basic serial approval process out of the box, but the workflow design is not highly customizable. For example, there is no way to design your own parallel workflows. Enterprises with more sophisticated needs may determine that Escenic’s workflow support is too basic.

Version 5.1 was the first to add support for full, uniform versioning of all content types, including digital assets. The system itself adds an API to enable this, and the “Revision History Module” takes care of storing all the versions. It seems questionable to have such a separated, heavyweight solution for versioning, but the approach has its merits; for one, it

relieves the core system of having to support the load of the inevitable bloat of archived content. The downside, is that this is still very much a work in progress.

Experience

Publishing

The default URL for an Escenic-based site looks like this:

hostname/section/subsection/article1234.ece

In previous versions it was necessary to map the proprietary “.ece” extension to more normal-looking URLs with some setup work — something that many customers never implemented. Version 5.4 came with “friendly URL” mapping, though the vendor hasn’t implemented this feature even on its own website.

Digital Marketing

Personalization services are in transition. In version 4.3, Escenic introduced the Profile Web Service API for writing modules that store user preferences and user account data to allow for personalization. This API has now been superseded by the new REST API, but Escenic hasn’t implemented a replacement for the profiling functionality yet. You should be aware that any advanced features in this respect (like the SMS validation some customers use) are built for the older version of Content Engine.

Analytics are available in an optional module, called the Analysis Reporter. Here you can view traffic statistics for your site (“publications”), detailed down to sections and individual articles. This module comes integrated with the Content Studio (the thick client).

Ancillary Services

The front-end web search uses Apache Lucene. It has interfaces for basic keyword search and a fielded “advanced search.” The functionality is relatively basic, but may be sufficient for your site; however, don’t assume extending the functionality will be easy. It might be more cost-effective to look into a third-party product that comes with all the trimmings and fittings you need.

Escenic has established partnerships with Leiki (for advertising and content recommendations) and Saplo (for text mining). Both of these are integrated into the product, but require an additional license fee. Leiki is primarily used for matching user behavior to both profile-based advertising and regular site content. Saplo is used to build semantic annotations around entities (e.g., people, places, companies, etc.) discovered within content. It should be noted with regard to Saplo that the product only supports Swedish and English languages; by comparison, the similar integration offered by Escenic’s rival Atex via the OEM’d version of

Experience	
Publishing	
Standards Adherence	<input type="checkbox"/>
Multichannel	<input type="checkbox"/>
Mobile	<input type="checkbox"/>
Digital Marketing	
Site & Campaign Analytics	<input type="checkbox"/>
Testing & Optimization	<input type="checkbox"/>
Segmentation & Personalization	<input type="checkbox"/>
Social Media Integration	<input type="checkbox"/>
Promotional Campaigns	<input type="checkbox"/>
Community & UCG	<input type="checkbox"/>
Workplace	
Collaboration & Networking	<input type="checkbox"/>
Dashboard	<input type="checkbox"/>
Ancillary	
Site Search	<input type="checkbox"/>
Online Forms	<input type="checkbox"/>
Module Ecosystem	<input type="checkbox"/>

Temis provides far greater language support, as well as a bigger array of extraction capabilities.

True to its strong foundation in the media space, Escenic has extended the product’s e-commerce capabilities, specifically to vend content. Editors can define sections as “chargeable.” Special templates will integrate with eSolutions’ Trusted Cash for payment.

You can obtain reader interaction on your site with discussion forums and blogs. Depending on your policies, you can decide to filter postings before or after they are published. Editors can also create basic polls, which can be deployed on the site. The basic modules for this come with the product; if you want more social software capabilities, and really build out a community, you’ll need the separately licensed Escenic Community Engine.

Vendor Intangibles

Support and engineering are based in Oslo, Norway. The company also has over 30 developers in Dhaka, Bangladesh.

According to licensees, support seems to work well and Escenic is good at setting expectations about how long bugs will take to investigate and fix. The company has worked closely with many of its customers to develop new features and even

entirely new modules. Several of them have a seat on the “customer advisory board,” which the vendor says is responsible for the direction development takes.

Escenic’s own consultants often participate in projects, but the company tends to use partners to implement projects rather than using its own people. (According to the vendor, it bases its business model on licensing revenue, not support or professional services.) The most experienced partners are available in the home market in Northern Europe, in particular Norway, Sweden, the Netherlands, Central Europe, and the U.K. Beyond this region, you’ll have a more difficult time finding experienced help. Escenic does claim to have an active user group, which you’ll surely want to investigate. The market for experienced Escenic professionals in most regions is very thin, meaning that attempting to find turn-key staff beyond Escenic’s own Professional Services operation can be difficult — if not impossible.

Licensing is complex, and while it starts at €30,000, fees typically go to €100,000, or more. Pricing is based on a combination of geography and industry, as well as the number of add-on modules required. A licensing deal is based on combination of page views per month, modules, and the number of editorial users. Asked about the highly complex nature of their licensing, Escenic claims it’s a matter of structuring licenses to suit customer needs — but you’ll do well to find out early on in a procurement process whether this will match your budget.

The long-anticipated version 5.0 was delayed for nearly two years. The company has several high-volume customers live on the 5.0.x branch, but if you’re planning for a high-traffic environment, do your own testing. A number of Escenic’s high-profile media customers are still using Version 4.x of the product — almost four years after version 5.0 was released. Migration from version 4.x to 5.x is far from trivial and many previously supported bespoke integrations require complete overhauls, which partly explains this version stasis for these

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

customers. When investigating references, we strongly advise you to ensure that the version you are researching is the same one that is used by the references.

Conclusion

Buyers in the media space might find Escenic a good fit, especially for complex online publishing scenarios. Escenic is a well-known, well-established player in its niche, with a loyal following in and beyond Scandinavia. In their home markets, neither CoreMedia, Oracle, nor OpenText are particularly strong in the media sector anymore, so Escenic has had the Nordics more or less to itself. That is not the case outside the region, where competition is stronger for media business.

Escenic is a powerful system for Multichannel Publishing and has shown that it can handle complex, highly interactive “Facebook” style sites, in a scalable manner — at least with older versions of the tool. Nevertheless, the workflow model is still basically a newspaper-publishing model involving simple serial approval steps.

Recent R&D efforts seem to be taking Escenic even further in the direction of purpose-built vertical applications for media customers, a process encouraged by its parent CCI. What this means is, if you’re not in the media business, you should take into account the fact that Escenic will probably continue to primarily tailor its R&D agenda to meet the needs of its media customers (which includes a long list of big accounts built up over many years), before it assumes other types of development.

In any case, remember that above all, this is a heavy-duty, enterprise-Java product that requires serious Java and XML skills to customize even some of its simpler components. If you want an agile solution, which allows you to quickly whip up a new implementation, throw together some modules and plugins and go, Escenic is not for you. Because let’s face it: When was the last time your daily newspaper had a major redesign?

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Simpler Product Vendors: Roll-Up Comparisons

- DotNetNuke: DotNetNuke
- Joomla!: Joomla!
- Kentico: Kentico CMS
- MODX: Revolution
- OmniUpdate: OU Campus
- OpenCms: OpenCms
- Salesforce: site.com
- TERMINALFOUR: Site Manager
- Telerik: Sitefinity
- Automattic, Inc.: WordPress

These are smaller, albeit typically quite well-established vendors that come to the marketplace with packaged tools at attractive prices. However, sometimes the tools are not as feature rich or as well tested as many of their “mainstream” competitors. Some, like TERMINALFOUR, are former regional vendors that have been able to expand beyond their regions and obtain a broader footprint — to varying degrees. Some of these vendors — like OmniUpdate — tend to focus on a limited number of verticals, tailoring their product features accordingly.

These packages tend to cater to do-it-yourself teams or smaller development shops, although the level of complexity varies among them. They tend to target use cases with fewer contributors and therefore, they generally have somewhat underdeveloped workflow and role management subsystems. At this tier, it is uncommon to see truly scalable installations and true separation of development and production environments. Nevertheless, the cost point is certainly attractive. Pricing in this category runs from US \$10–60,000+.

Specsheet Summary: Simpler Products

	DotNetNuke	Joomla!	Kentico	MODX
Geography	Global, with the largest footprint in North America and Europe	Global	Global partner network, HQ in the Czech Republic	Global, with emphasis on North America
What's New	<ul style="list-style-type: none"> Acquisition of Snowcovered, which is now the DNN application store Beginning to integrate the recently acquired Active Modules for social community forums German, Dutch, Spanish, and French language packs added OpenID support for user authentication SharePoint 2007/2010 connector in Enterprise Edition 	<ul style="list-style-type: none"> Multi-database support with the addition of MS SQL Server Further enhancements to upgrade process Search now based on a separate index, rather than database 	<ul style="list-style-type: none"> E-Marketing edition adds a plethora of new functionality Cloud version of the product now available Visual workflow designer and more formalized staging services included in some versions 	<ul style="list-style-type: none"> "Media Sources" integrates with cloud storage service
Strengths	<ul style="list-style-type: none"> Built off well-established .NET 3.5/4.0 Framework Broadly extensible via module and provider structures Offers impressive "portal-in-a-box" offering; additional modules are available at app store Vibrant community of consultants, skin designers, module developers — both commercial and open source Native blog is feature rich and well thought out for public blogging System can access multiple authentication providers Commercial options include the Professional and Enterprise Edition 	<ul style="list-style-type: none"> Sites of low to moderate complexity can be rolled out quickly Friendly, GPL 2 license Can generate standards-compliant XHTML markup and human-readable, search-engine-friendly URLs User interfaces are Spartan but friendly-feeling, easy to navigate Supports wide range of authentication services: LDAP, OpenID, Gmail Authentication, or Joomla!'s own authentication Modules can be reused and tailored to deliver customized content on each page Product benefits from a large, growing, vibrant ecosystem Many third-party extensions available Developers with the skills needed to customize Joomla! (i.e., knowledge of PHP, SQL, XHTML, CSS) are not hard to find 	<ul style="list-style-type: none"> Unusually feature-rich product, with a plethora of modules at a relatively modest price point Relatively easy to customize and tweak for simpler needs Comparatively good support and documentation — a rarity at this tier E-marketing features might be comprehensive enough for some small enterprises Adheres to modern .NET standards 	<ul style="list-style-type: none"> Flexible, developer-friendly platform is more modern than Joomla! Employs established PHP toolsets, such as xPDO for object-relational mapping and Smarty templating Small design agencies find the learning curve is easier than Drupal's Page-oriented approach works well for sites with simpler content models and limited content reuse Increasing availability of commercial support for integrators and end users

(Continued)	DotNetNuke	Joomla!	Kentico	MODX
Weaknesses	<ul style="list-style-type: none"> • Extraordinary dearth of native management services (e.g., weak workflow, versioning, taxonomies, and Office integration) • Modules are difficult to develop and also introduce upgrade uncertainty • Role management is coarse grained and likely ill-suited to delegate responsibilities on large sites • Managing multiple portals off one instance can become problematic • Skinning approach to templating feels unduly complex and technical compared to most other systems • Crude staging, promotion, and configuration services make DNN an unlikely candidate for multi-environment enterprise deployments • Generates non-standard URLs by default • Community is broad, but not deep; lacks support of large consulting firms and is sparse in some regions 	<ul style="list-style-type: none"> • No true workflow capability • Poor support for content reuse and re-purposing • Lack of support for custom content types make it an unlikely candidate for all but the simplest of sites • No support for multi-server environments or multisite management • Low levels of community support for SQL Server option at this time • Very tightly coupled, single-point-of-failure design makes it a poor choice for enterprise-wide deployments • Persistent security concerns and comparatively less savvy customer base renders Joomla! a magnet for possible attackers • Surprisingly uneven quality control for an open source project; frequent patching and attention required • Recent versions have had very small support windows • Community is somewhat fractious and anarchic • Joomla! users will see shorter version support windows, in a likely pyrrhic attempt to push faster upgrade behavior 	<ul style="list-style-type: none"> • Breadth rather than depth of features means some functionality not fully baked when released (e.g., mobile preview) • Crowded UI may prove overly complex for casual contributors • Repository and audit services still lag in some areas, reducing enterprise value • Lack of integration among e-marketing functions reduces long-run value of those add-on modules • Bolted-on staging approach likely won't work for larger, more configuration-management-conscious customers • Overly aggressive sales and marketing tactics may prove off-putting • Not well suited for more than very basic globalization scenarios 	<ul style="list-style-type: none"> • Localization services rely heavily on community contributions, resulting in inconsistent quality and limited coverage • Weak directory integration could cause problems for larger implementations • Still a development platform most suited for customers with strong internal IT depts. or relationships with experienced system integrators • No real multisite management services or content inheritance model • Rudimentary workflow capabilities and an absence of versioning services may limit uptake in larger enterprises • User interface is designed for power users and its lack of in-context editing could intimidate casual contributors • Dearth of experienced integrators may reduce your implementation support choices • Community's efforts are divided between two overlapping but incompatible products
Potential Fit	Community-oriented Site, Informational Site, Microsites & Landing	Simple Informational, Community Oriented, Basic Digital Marketing	Community-Oriented Site, Informational Site, Basic Marketing	Informational, Microsites
Unlikely Fit	Global Enterprise, Advanced Marketing Portal, Multichannel Publishing, Digital Workplace	Mid-range and Complex Scenarios	Global/Enterprise Digital Workplace, Multichannel Publishing, Ultra-Large Single Site, SMB/Departmental Digital Workplace	Any Complex or Digital Workplace Scenario
Compare To	Drupal, Plone, Microsoft, Ektron, Kentico, WordPress	Drupal, eZ Systems, TYPO3	Ektron, EPiServer, Telerik, Sitecore, DotNetNuke	WordPress, Plone, Joomla!, Drupal

(Continued)	DotNetNuke	Joomla!	Kentico	MODX
Operating Systems	Windows	Windows, Linux/UNIX	Microsoft Windows	Linux x86 / x86-64 Bit, Windows Server, OS X
Repository	Database: SQL Server	Database: MySQL, Microsoft SQL Server	Database: MS SQL	MySQL, Microsoft SQL Server
Client	Browser	Browser: IE and Firefox, TinyMCE WYSIWYG editor for rich text editing	Browser: IE, Firefox, and Safari — Chrome not fully supported	Browser
App Platform	.NET	LAMP	.NET	PHP/Apache
Licensing	Open source (BSD-style), commercial license starts at \$2,500/year/instance	Open Source (GPL 2)	Based on functional options, for one website base license at \$1,999, to one server ultimate license for unlimited sites at \$4,499, full e-marketing options at \$14,999 (or \$999p/m for the cloud version)	GNU GPL Open Source License
Ownership	Privately held	N/A	Privately held	Privately held

	OmniUpdate	OpenCms	Salesforce
Geography	USA and Canada	Global, with emphasis in Europe, particularly Germany	Global
What's New	<ul style="list-style-type: none"> Automatic XML sitemap creation Accessibility testing module "Multi-Target Publish" deployment support 	<ul style="list-style-type: none"> Improvements to page authoring and in-context editing 	<ul style="list-style-type: none"> site.com evolves from pre-existing Siteforce product
Strengths	<ul style="list-style-type: none"> Available most commonly as Saas Focused on higher education, with many references in that sector Decoupled architecture and page-based orientation will appeal to universities with simple websites Good accessibility controls and multi-browser preview services Company puts new features in production on a frequent basis Well-regarded user group adds good value to the offering, especially for higher-education customers The UI and product features seem optimized for infrequent, non-technical contributors 	<ul style="list-style-type: none"> Core source code and binaries are truly free Strong repository services and excellent search A decent ecosystem has evolved, particularly in the DACH region Product development proceeds at a steady, reasonable pace Bugs are exposed in an openly-available bug tracker You can purchase optional modules for enterprise functionality (e.g., replication, LDAP) Repository supports CMIS 	<ul style="list-style-type: none"> Low-complexity sites can be rolled out quickly Decent Salesforce integration means relatively transparent CRM data interchange Editors can use Salesforce Chatter to collaborate on content creation Customers have access to Salesforce's own CDN Company promises updates three times a year
Weaknesses	<ul style="list-style-type: none"> Workflow supports simple approvals only Minimal support for true multisite management Dearth of interactive services — favors mostly static publishing Not well suited for component content reuse Comparatively poor support for managing images and documents; it's not a good fit for most intranets May be experiencing growing pains with a comparatively small staff of only around 60 employees User interface is outdated and is only available in English On-premise version is essentially an appliance requiring remote administration by OmniUpdate 	<ul style="list-style-type: none"> Quality testing is done partly by Alkacon Software GmbH, partly by the community, and partly by customers already in production Dearth of asset management services makes it a poor fit for media-rich sites Default interface is geared toward power users Almost complete lack of digital marketing services rules out many modern web publishing scenarios Clustering and replication require additional commercial modules Add-on modules can be difficult to install and manage as there is no formal third-party module validation or testing process Commercial support options are limited Alkacon itself competes with erstwhile consulting partners, which may limit growth of the community 	<ul style="list-style-type: none"> Extraordinary lack of library services (versioning, version control) makes system more of an authoring service than a management platform Absence of workflow and overly simplistic access model make the platform ill-suited for all but the smallest editorial teams Employs a proprietary rich text editor, which could prove finicky Lack of native services for content reuse or multisite management render the platform cumbersome for larger enterprises Development environment is dated and arcane Salesforce does not like to provide service level agreements
Potential Fit	Informational Sites, Community Oriented	Informational Site, SMB/Departmental Digital Workplace	Simple Informational, Microsites
Unlikely Fit	Digital Workplace, Global Enterprise, Ultra-Large Single Sites	Community-Oriented Site, Basic Digital Marketing, Complex Scenarios	All Mid-range and Complex Scenarios

(Continued)	OmniUpdate	OpenCms	Salesforce
Compare To	CrownPeak, Upland Software, Hannon Hill, Ingeniux, TERMINALFOUR	Plone, Hippo, Magnolia	WordPress, CrownPeak, Upland Software
Operating Systems	Linux, Solaris, Windows for on-premise Enterprise Server	Windows, Linux (Red Hat, Debian, SUSE)	N/A
Repository	On-premise bundles Sybase; also supports MySQL & MS SQL Servers	Databases: Any JDBC-capable database	N/A
Client	IE, Firefox, Safari, Chrome in Windows, Mac, and Linux	Browser: IE, Firefox, Safari, Chrome	Browser
App Platform	Software as a Service; optional Java, on-premise appliance with embedded database and application server	Java (6 or 7). Prefers Tomcat but runs on multiple servlet containers	Force.com
Licensing	Subscription-based, starting at US \$21,000/year	Core: Open source — LGPL, plus optional commercial modules	Software as a Service: From \$19,500/year for a single site with single user access Additional users: \$1,500/year per Publisher, \$240/year per Contributor
Ownership	Privately held	Alkacon Software — GmbH owns most copyrights	Public (NYSE: CRM)

	Telerik: Sitefinity	TERMINALFOUR	WordPress
Geography	Primarily North America with some UK, Netherlands, and Australia	Mostly Europe and North America	Global
What's New	<ul style="list-style-type: none"> • New Site Sync feature provides improved content deployment options • New email marketing tools have some segmentation functionality • Visual Studio integration and ASP.NET MVC support 	<ul style="list-style-type: none"> • Automated migration and import tool for various CMS packages • Reporting integration with Google Analytics 	<ul style="list-style-type: none"> • New, upscale hosting and support options from Automattic and others • Roadmap published for major 4.0 release • Noticeable uptick in media customers

(Continued)	Telerik: Sitefinity	TERMINALFOUR	WordPress
Strengths	<ul style="list-style-type: none"> • Exceptionally low starting price • Cross-browser compatible WYSIWYG editor • Comes with a slew of additional application modules • Very shallow learning curve for .NET developers • Functional administration interface (especially for this price range) that's comparatively simple to learn • Company upgrades functionality on a monthly basis • Nice features for feed aggregation and syndication 	<ul style="list-style-type: none"> • New content types and templates are relatively easy to create • Supports unusually sophisticated workflows • Live previewing of PHP, ASP, and JSP pages • Product produces short, friendly URLs (that can be manually edited) • Unusually strong support for accessibility standards • Very active and open customer community with emphasis on higher-education institutions in UK, Ireland, North America • Company appears to be responsive to feature requests • High-value offering for smaller websites with modest requirements • Interfaces are localized into Welsh, Irish, French, German, & Spanish 	<ul style="list-style-type: none"> • Well suited to Simpler scenarios • Broad and deep open source community, which has localized the platform into more than 70 languages • Unusually simple product to install • One of the few products that is almost universally liked by all contributors • Many third-party modules available • Extraordinarily diverse hosting/cloud options for nearly any budget • Growing traction among media firms • Excellent fit as "second" CMS for enterprise microsities
Weaknesses	<ul style="list-style-type: none"> • Relatively immature product in terms of feature richness • Sub-standard repository and information lifecycle services • Coupled management and delivery environments may not appeal for some IT departments • Sitefinity has seen potentially serious back-end performance issues, especially with large sites • Weak user management and reporting features • Weak native services for separating development, staging, and production environments reduces enterprise applicability • Templating requires developers • Company does not offer real-time, 24/7 tech support 	<ul style="list-style-type: none"> • Company knowledge of markets outside of education, membership organizations, government, or financial services is limited • Page-based orientation is not ideal for content reuse scenarios • Surprisingly few native features for multisite management • Although possible to extend, it's not a development platform compared to other offerings and is not ideal for complex scenarios • Peculiar, content-oriented licensing model punishes larger websites • Company appears heavily focused on services and runs a bit like a consulting operation • With a dearth of knowledgeable consultancies in the tool, customers depend on the vendor itself for professional services 	<ul style="list-style-type: none"> • Rudimentary user management and limited access control reduces suitability in many enterprise environments • Has no native content modeling: still requires development or plugin for anything beyond "page" and "post" types of content • Many key enterprise features are not in the core, and must be realized through third-party modules of variable provenance • No at-a-glance view of the page hierarchy and lack of in-context editing limits manageability of larger sites • Irregular new releases in quick succession have administrators tied up in a continuous update process • Popularity and openness of the software leaves sites persistently vulnerable to malicious attacks • Comparatively more difficult to integrate with other enterprise systems
Potential Fit	Informational, Community-Oriented Site, Microsites & Landing Pages	Microsites and Landing Pages, Informational Sites, structurally simple Ultra-Large Sites	Informational, Microsites & Landing Pages
Unlikely Fit	Advanced Marketing Portal, Global Enterprise, Multichannel Publishing	Community-Oriented Site, Global/Enterprise Digital Workplace, Advanced Marketing Portal, Global Enterprise	Complex Scenarios

(Continued)	Telerik: Sitefinity	TERMINALFOUR	WordPress
Compare To	Ektron, Sitecore, EPiServer, Kentico, DotNetNuke	Hannon Hill, Percussion, OmniUpdate	Joomla, Drupal, DotNetNuke, TERMINALFOUR
Operating Systems	Windows Server	AIX, Windows (including Vista), Solaris, Linux	Windows or Linux
Repository	MS SQL Server, MySQL, and Oracle	Databases: Microsoft SQL, MySQL, Oracle, PostgreSQL	MySQL
Client	Browser (Silverlight required for analytics module)	IE, Firefox, Safari, Opera, on Windows/Vista, Mac, Linux, Solaris	Browser
App Platform	ASP.NET	Java: WebSphere, Oracle WebLogic and OAS, Tomcat, JBoss	PHP
Licensing	Commercial license starts at \$499 per domain	Based on number of content items; €10-50,000 plus 22% for support	Open Source (GNU GPL)
Ownership	Privately held	Privately held mostly by company principals with some equity financing	Privately held with 250 employees

Category Summary: Simpler Products

Phase / Attribute	DotNetNuke	Joomla!	Kentico	MODX	OmniUpdate	OpenCms	Salesforce	Telerik Sitefinity	TERMINALFOUR	WordPress
Technology										
Technical Administration & Security										
Threat Prevalence										
Authentication & Authorization										
System Reporting										
Multisite Management										
Cloud Services										
Development										
Configuration & Customization										
Integration & Extension										
Content Modeling										
Templating										
Performance										
Back-end Performance										
Site Caching & Delivery										
Content										
Contributor Experience										
Overall Usability										
UI Accessibility										
Contributing Content										
Authoring & Transformation										
Tagging & Taxonomy										
Content Reuse										
Media & Document Management										
Repository Services										
Content Lifecycle										
Workflow										
Globalization										
Archiving & Compliance										

Phase / Attribute	DotNetNuke	Joomla!	Kentico	MODX	OmniUpdate	OpenCms	Salesforce	Telerik Sitefinity	TERMINALFOUR	WordPress
Experience										
Publishing										
Standards Adherence										
Multichannel										
Mobile										
Digital Marketing										
Site & Campaign Analytics										
Testing & Optimization										
Segmentation & Personalization										
Social Media Integration										
Promotional Campaigns										
Community & UCG										
Workplace										
Collaboration & Networking										
Dashboard										
Ancillary										
Site Search										
Online Forms										
Module Ecosystem										
Vendor Intangibles										
Vendor Services										
Vendor Professional Services										
Channel Partner Services										
Support & Community										
Strategy & Roadmap										
Viability & Stability										

Key	
	Product does not provide this feature in any meaningful way.
	Product provides this feature, but is not as mature as its rivals
	Product provides this feature
	Product excels at this feature, <i>relative to other products in the same category</i>
	Product masters this feature, <i>relative to other products in the same category</i>

Scenario Fits Summary: Simpler Products

Phase / Attribute	DotNetNuke	Joomla!	Kentico	MODX	OmniUpdate	OpenCms	Salesforce	Telerik Sitefinity	TERMINALFOUR	WordPress
Simpler Site										
Informational										
Microsites & Landing										
Mid-Range										
Basic Digital Marketing										
Mobile Site										
Community Oriented										
Complex Site										
Advanced Marketing Portal										
Global Enterprise										
Multichannel Publishing										
Ultra-Large Single										
Digital Workplace										
SMB/Departmental										
Global/Enterprise										

Key	
	Product does not provide this feature in any meaningful way.
	Product provides this feature, but is not as mature as its rivals
	Product provides this feature
	Product excels at this feature, <i>relative to other products in the same category</i>
	Product masters this feature, <i>relative to other products in the same category</i>

DotNetNuke: DotNetNuke

dotnetnuke.com

Vendor at a Glance

Specsheet	DotNetNuke: DotNetNuke 6.2 Summary
Geography	Global, with the largest footprint in North America and Europe
What's New	<ul style="list-style-type: none"> • Acquisition of Snowcovered, which is now the DNN application store • Beginning to integrate the recently acquired Active Modules for social community forums • German, Dutch, Spanish, and French language packs added • OpenID support for user authentication • SharePoint 2007/2010 connector in Enterprise Edition
Strengths	<ul style="list-style-type: none"> • Built off well-established .NET 3.5/4.0 Framework • Broadly extensible via module and provider structures • Offers impressive “portal-in-a-box” offering; additional modules are available at app store • Vibrant community of consultants, skin designers, module developers — both commercial and open source • Native blog is feature rich and well thought out for public blogging • System can access multiple authentication providers • Commercial options include the Professional and Enterprise Edition
Weaknesses	<ul style="list-style-type: none"> • Extraordinary dearth of native management services (e.g., weak workflow, versioning, taxonomies, and Office integration) • Modules are difficult to develop and also introduce upgrade uncertainty • Role management is coarse grained and likely ill-suited to delegate responsibilities on large sites • Managing multiple portals off one instance can become problematic • Skinning approach to templating feels unduly complex and technical compared to most other systems • Crude staging, promotion, and configuration services make DNN an unlikely candidate for multi-environment enterprise deployments • Generates non-standard URLs by default • Community is broad, but not deep; lacks support of large consulting firms and is sparse in some regions
Potential Fit	Community-oriented Site, Informational Site, Microsites & Landing
Unlikely Fit	Global Enterprise, Advanced Marketing Portal, Multichannel Publishing, Digital Workplace
Compare To	Drupal, Plone, Microsoft, Ektron, Kentico, WordPress
Operating System	Windows
Repository	Database: SQL Server

Specsheet	DotNetNuke: DotNetNuke 6.2 Summary
Client	Browser
App Platform	.NET
Licensing	Open source (BSD-style), commercial license starts at \$2,500/year/instance
Ownership	Privately held

Summary

DotNetNuke (DNN) might seem at first blush like an oxymoron: an active open-source project in the Microsoft/.NET world. Well, the project is real, and grew quite large and active, responding to the needs of those who want to stay in the Redmond world, but don't have the budgets for SharePoint. After the project's January, 2009 move from SourceForge, it is now one of the flagships of Microsoft's CodePlex open source development site.

Much of the enthusiasm around DNN centers on its concept of pluggable "modules." Powerful, but not easy to create, commercial suppliers of all stripes have stepped in to offer open source or fee-based modules to extend or enhance the platform. These give DNN a portal flavor, but like Drupal, it's really more of a community site in a box, rather than an integration platform. DNN customers should exhibit caution when selecting modules and put aside some time for research, since modules tend to vary greatly in terms of quality and ongoing support by the initial creator.

Our chief knock against DNN is its almost breathtakingly weak content production services, ranging from metadata to basic repository services. You can publish new content quite easily in DNN, but not easily *manage* content in a serious enterprise setting. Findability is another major turn-off: DNN's native search services are weak and the platform lacks simple navigability — even for power users.

That might not matter if you're looking for an open source (but probably not quite free) alternative to SharePoint. Still, you're likely to find more mature and deeper content management services with the likes of Plone or Drupal — if you can get over the fact that neither run on .NET. Among commercial competitors, you may want to compare DNN to Sitecore and Ektron — both pure .NET products.

Over time, the platform may improve, as the DotNetNuke Corporation elaborates a more enterprise-y (and hardened) Professional and Enterprise Editions, which offer some support and extended feature sets. These come at an annual fee of \$2,500 and \$5,000 respectively per

Scenario Fits	
Simpler Site	
Informational	<input checked="" type="checkbox"/>
Microsites & Landing	<input checked="" type="checkbox"/>
Mid-Range	
Basic Digital Marketing	<input checked="" type="checkbox"/>
Mobile Site	<input checked="" type="checkbox"/>
Community Oriented	<input checked="" type="checkbox"/>
Complex Site	
Advanced Marketing Portal	<input type="checkbox"/>
Global Enterprise	<input type="checkbox"/>
Multichannel Publishing	<input type="checkbox"/>
Ultra-Large Single	<input type="checkbox"/>
Digital Workplace	
SMB/Departmental	<input type="checkbox"/>
Global/Enterprise	<input type="checkbox"/>

DNN instance. Recently, DotNetNuke also started selling “Elite Support” at \$3,000 per year, per instance. This is useful if you want someone from DNN support on the phone; the rest of support models are only online based.

Introduction

Like Plone, DotNetNuke (DNN) is part portal, part content management framework, but it is all based on Microsoft’s .NET technology. The original codebase emerged in 2002 from an early .NET reference project developed by Shaun Walker, called “IBuySpy Workshop.” Walker changed the name to DotNetNuke to take advantage of the (then) popularity of the “Nuke” moniker, which implied a kind of open, community-oriented platform.

In September 2006, Walker and three partners cofounded the DotNetNuke Corporation, which owns the copyright to the name and codebase, and manages the core platform rather tightly. At the same time, a vigorous (and largely commercial) ecosystem emerged around the platform, which had substantial growth over the subsequent years.

The company is based in San Mateo, California, with an engineering office near Vancouver, Canada, and a recently opened European base in Amsterdam. A little more than 60 full-time employees are complemented by efforts from a number of remote workers around the globe.

Since its inception, the platform underwent various upgrades through 3.x, 4.x, and 5.x cycles, most of them incremental rather than monumental. It remains at its core a kind of .NET 3.5/4.0Framework, with various pluggable “modules” built on top.

About a dozen basic modules are included with the core DotNetNuke distribution. You can find a myriad of additional third-party modules, as well as “skins” (templates in DNN’s jargon), developed by both the open source community and proprietary commercial developers. “The Store” (formerly Snowcovered.com, acquired and brought in-house in 2011) is the vendor’s application marketplace, and is one of the locations where you can find and buy the coveted skins and modules (like web forms, or calendars) for an average of \$100 each. Since DNN documentation remains quite thin and occasionally outdated, you can also purchase a variety of user guides. Customers advise buying these third-party publications when seeking help in navigating the DNN waters.

DotNetNuke’s Professional Edition version 5.6.2 was released in March 2011, and provided various enhancements to the security model and performance. Version 6.2, released in May 2012, primarily added the social/community elements from the acquisition of Active Modules (a set of add-on modules, formerly available via DotNetNuke’s store). A SharePoint 2010 connector was also added for “Enterprise Edition” product subscribers.

Feeling the heat of criticism on the product’s architectural shortcomings from enterprise customers, DotNetNuke Corporation introduced a Staging environment in its “Enterprise Edition,” which also ships with a few other distinguishing features. This edition also includes “Advanced Content Approval” (which is still a very basic workflow, but the community edition only has “single stage approval”), and more granular permissions.

Technology

Technical Administration & Security

DNN requires Windows (and the .NET 3.5 or 4.0 Framework), IIS webserver, and a database. By default, it assumes SQL Server, though it's possible to employ SQL Server Express for development.

DNN's architecture allows you to connect to any relational database management system, but today it only officially supports SQL Server.

As with many other tools, installation can get a bit tricky, but the latest versions include a helpful wizard. At 250,000 lines of code, this is not a slim package.

DNN is a "coupled" architecture, and one gets the sense that the typical installation runs on one box. In fact, the package proudly allows you to intermingle visitors and contributor services to the extent that your security model allows. However, many customers are going beyond the one-box approach, even though DotNetNuke says you can run as many sites from one instance as you want, via IIS configurations. In reality, you will want to separate public-facing websites from intranets via separate web servers, each with its own SQL Server instance. You'll probably want to introduce a separate development server into this architecture.

Technology	
Technical Administration & Security	
Threat Prevalence	◐
Authentication & Authorization	◐
System Reporting	○
Multisite Management	○
Cloud Services	◐
Development	
Configuration & Customization	◐
Integration & Extension	◐
Content Modeling	◐
Templating	◐
Performance	
Back-end Performance	◐
Site Caching & Delivery	◐

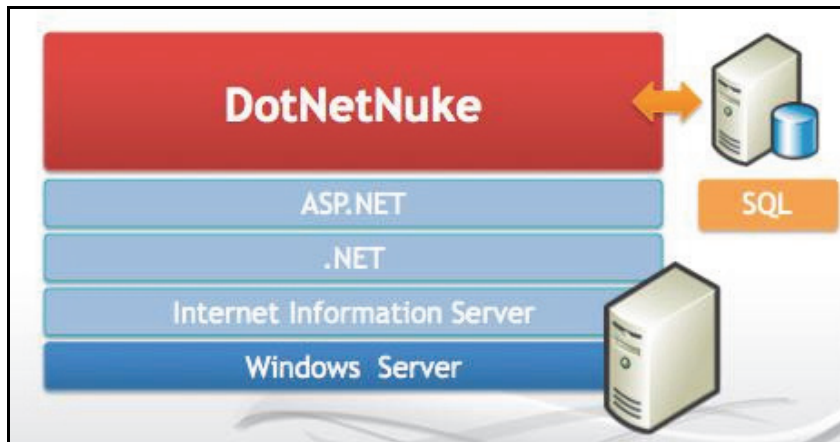


Figure 350. A typical architectural setup for a DNN implementation runs on a standard Microsoft stack. While the vendor will tout its multi-tenancy and multisite management capabilities, you'll want to investigate this area carefully if your web presence is not simplistic. (Source: DotNetNuke Corporation.)

Introduced for Enterprise Edition subscribers in 2011, the Content Staging service aims to bridge the gap for more complex, enterprise-y installs, where you need de-coupling of some environments, including physically separate boxes. This feature works in the traditional sense,

allowing you to preview and test content changes before it's published to a live production website.

To date, the long-lasting lack of content staging and promotion capabilities have been considerable drawbacks for DotNetNuke. Even now, don't expect much sophistication here yet, since the mechanisms for content promotion remain in their infancy. You should also know that Content Staging is not well suited for any code promotions. A traditional, separate development server is still the safest way to go for testing and QA before deploying new modules or code to a production environment. For \$5,000 per instance, per year for the Enterprise Edition, the only differentiator is this (suspect) Content Staging function.

With respect to access control, DNN has roles, but not groups. Super users can create user accounts for any person, or they can enable site visitors to register themselves for a user account. They can also define permissions against individual users, as well as to roles.

- **Module Editors:** Can edit one or more modules, but likely not an entire page
- **Page Editors:** Can edit one or more pages and edit module settings on their pages, as well as add child pages
- **Administrator:** Can manage all page and module content on a single portal (site)
- **Host Account:** A quasi-administrator, this account ships with DNN by default and allows "super-user" access to the entire installation (all portals). For each portal (site), there is an administrator role and a registered user role.

In reviewing the roles, you can see it is not a particularly fine-grained system, and tends to assume that editors are power users, as well. As usual, there are third-party modules you can find that will enable more delegated administration, but they tend to be isolated solutions with limited community support.

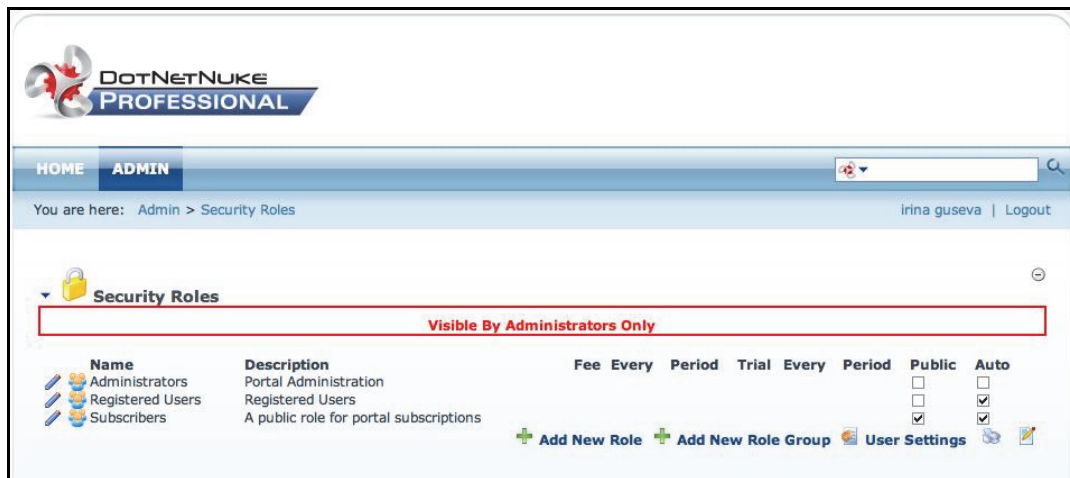


Figure 351. An example implementation of security roles in DotNetNuke. If you're looking for granularity in user and role management, DNN may not be the best fit.

Going from version to version of DNN, you will see more confusing restrictions managing the security model. In the Community Edition, permissions can be set on pages, modules, and folders. However, the set of permissions is restricted to View and Edit rights. In the

Professional and Enterprise Editions, there are more granular rights and permissions available, including Add, Modify, Delete, Copy, Import, Export, etc.

Like Drupal, DNN assumes that visitors will also serve as contributors, and the system is architected for provisioning (even self-provisioning, if you allow) large numbers of external contributors.

- **All Users:** All unauthenticated and authenticated users
- **Unauthenticated Users:** Anonymous visitors who are unregistered or not logged in
- **Registered Users:** Can manage their User Account online, and may belong to one or more security roles

On the plus side, DNN offers several ways to authenticate contributors and visitors. You can employ Active Directory, though you need to configure AD and IIS carefully, or it won't work. DNN also has a notion of custom authentication providers, which enable you to tap other identification repositories, including standard databases. Version 6.2 added the ability for users to be authenticated using OpenID so pre-existing credentials from Facebook, Twitter, Google, or Microsoft accounts can be used. While this certainly reduces the amount of friction required for users managing multiple accounts, the danger of this lack of separation between site users and site administrators is somewhat compounded by the potential security breach caused by OpenID. In short, if an account were compromised and unnoticed outside of the system, it could wreak havoc if maliciously obtained.

Reporting is only available via third-party modules.

Like other portals, DNN is organized hierarchically: From parent portal, to one or more child portals, to individual pages (which can have child pages, each with their own modules). A key consideration here is default support for multiple portals. On the one hand, this is a potential strength, initially adopted by some hosting companies to put multiple customers on one DNN installation. You can also set site subdirectories as different "portals," which allows you to skin them differently and — perhaps more importantly — to enforce different security and access control settings in a manageable way.

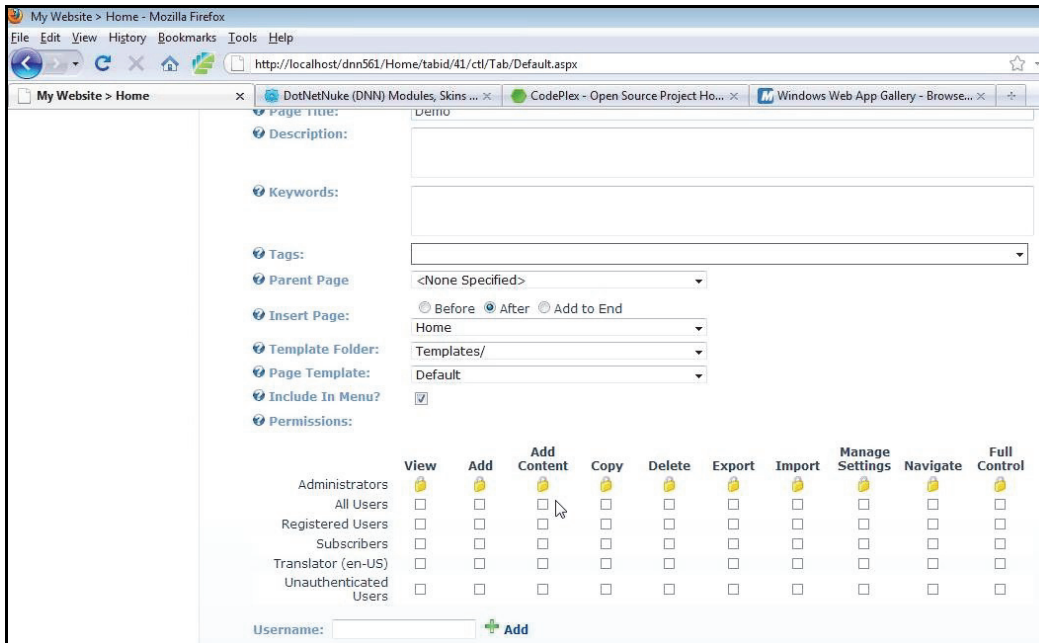


Figure 352. Security settings management on a page level in DNN.

On the whole, the community seems to be getting weary of the multi-portal assumptions built into the package. One frustrated developer says, “DotNetNuke spends an inordinate amount of energy supporting the ability to make multiple portals.” First, not all third-party modules will work in a multi-portal environment. One customer offered a typical complaint, “The module didn’t want to play nicely with child portals,” resulting in the additional effort by the organization to fix the issues in a back-and-forth with the module’s original developer. You cannot restore a single portal from a crash — you must restore the entire database, which contains all content. In some cases, a faulty module can take down the entire collection of portals.

Development

With respect to customization and integration, DNN revolves around the concepts of “modules.” These are components that resemble portlets in the Java portal world, or Web Parts in the SharePoint world, but they have a very specific structure and database bindings. They require that you develop specialized “.dnn” files to populate XML configurations (alternatively, you can use DNN’s Packaging Wizard, which generates the .dnn manifest file automatically if you have properly configured the module). Then you create a “code-behind” .ascx file (with suitable ASPX coding), which, in turn, likely calls a compiled file that stores your logic (written in some .NET language like VB.NET or C#). For more complex modules, you also need to create some stored procedures in the appropriate database table. HTML is generated when DNN processes the skin control (*.ascx), inserting page content. After the page is filled with content, the control passes to the ASP.NET engine, which renders the page and substitutes ASP.NET tags with proper HTML code.

While you easily could find cleaner approaches among DNN’s competitors, this framework will be familiar to most .NET developers.

As with Drupal and other modular, open source platforms, when you upgrade the core, you may need to upgrade various dependent modules for them to remain compatible (and vice-versa). The DNN Corporation is good about tracking this for the modules that it supports, but for everything else, it’s up to you to track. Many customers find this is a burden, and (without extensive research) it’s hard to rely on certain modules coming out of the community.

Note that most configurations live on the file system, and DNN does not have a way to manage them internally. That means you need to rely on your own configuration management system for all code-based changes, and you must track other browser-based settings carefully. There are no native roll-backs. Like SharePoint, this is a system that can punish novice developers.

There isn’t really a notion of content types here. If you want a new content type, you have to create a new module to support it; that means some serious coding. To be sure, the default editorial modules — Contacts, Chat, Events, FAQs, Gallery, Help, Reports, What’s New, FCKeditor, Blogs, Wikis, Surveys, and more — provide a nice starter set for a community site. Because of this, you would not want to use DNN for highly structured content. Even though DNN is not page, but module oriented (in theory allowing you to reuse and extend modules across different sites), the platform is not designed for granular content reuse.

Templates are driven by DNN “skins.” These represent a collection of files that reside on the file system — and therefore, you must manage separately, outside the system. The main file is a quasi-HTML file that contains some DNN-specific tags and tokens. You invoke a parsing routine that converts it to an .ascx file (which replaces tags and tokens with .NET code); this then is loaded into the default .aspx display file the next time it’s called. Once installed, you can apply skins at the Portal or page level, and it’s relatively easy for page managers to swap them in and out via a browser interface.

There is a thriving commercial marketplace for skins, not the least of which because the default DNN skin only supports a simplistic, three-column layout. The skinning approach is

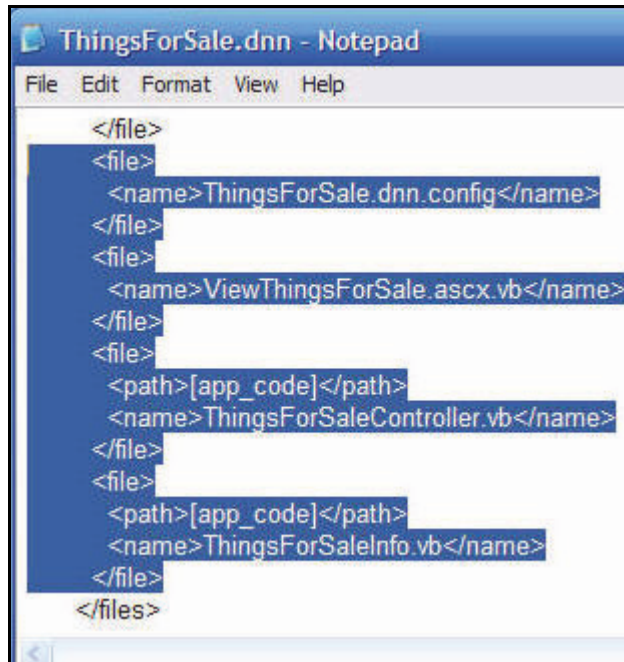


Figure 353. A sample .dnn module file for a component that’s designed to show a list of items for sale. Note that it references four other files for this one component.

quite flexible, and some designers have created some quite attractive-looking portals with DNN, but you need to be careful here, because some skins are quite heavy (300+kb), inaccessible, and IE specific.

Perhaps more importantly, creating and modifying a skin typically requires developer expertise, though the best skins use CSS, and therefore can be modified to some extent by a competent designer. Entire books have been written on skinning DNN. Moreover, the skin may not have much effect on how its individual modules are displayed (depending on the module), you may need to muck around a bit to modify the complete experience. As a practical matter, few DNN site owners seem to modify module appearance, which is dictated by skin-like “container” files. This tends to give DNN sites a particular look, even if the overall wrappers and structural layouts are infinitely variable.

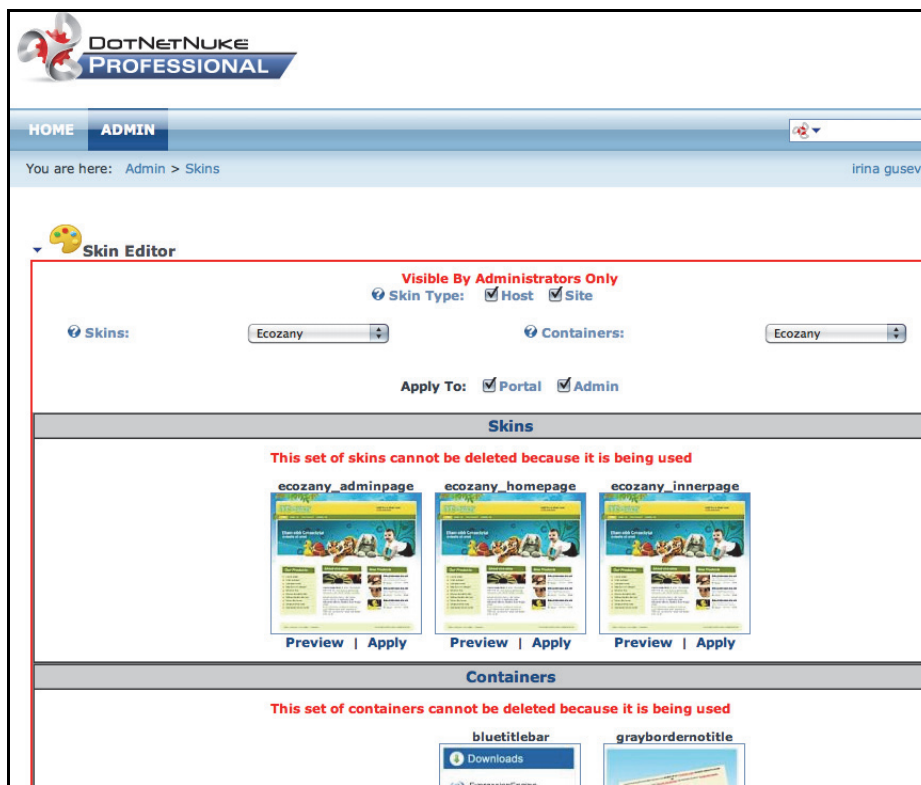


Figure 354. DNN Skin Editor allows you to manage and preview skins and containers.

There is some debate in the DNN community about whether skinning is superior to the .NET 3.0 / SharePoint approach of Master Pages. It’s likely a moot point, since skins are here to stay in DNN. In versions 5.x, skins are implemented as “extensions,” rather than version 4’s straightforward ZIP files. This means deployment should be more coherent (e.g., skins now also have version numbers), but designing a new skin has become more of a developer’s job, as well.

Using a nifty “iframe” module that comes with the default package, you can display HTML from other sites. Note that this is not an actual integration — your site will be blissfully

unaware of what goes on inside the iframe, and in general, it may be difficult to develop a site using them that will conform to accessibility guidelines.

Performance

Performance is a perpetual issue with all dynamic delivery systems, and DNN is no exception. Event and traffic logging can become a resource hog on even medium-sized sites and need to be monitored, with logs rolled regularly. Hosting multiple portals off one install can magnify database size and slow query times dramatically. You can cache to disk individual HTML fragments, which are the outputs for each module on a page, based on specific expiration/refresh attributes. You’ll want to make sure to configure these accordingly for maximum performance. Caching setup and configuration is one of the most common technical challenges in this tier of CMS vendors. DNN also comes with some optional compression utilities, but they are reputedly tricky to configure and can be buggy.

Release code quality has also affected the overall system performance with DotNetNuke. Several releases that went out in 2010 had serious quality-control issues. Reportedly, customer complaints prompted DNN to invest more in their QA team, bringing it to seven full-time employees (about half the size of DotNetNuke’s development group). The vendor says it’s focusing equally on both manual and automated testing, but it remains to be seen whether this focus on quality control will remain.

Another possible complication is DNN’s decision to move the core of the CMS to C# from VB. While not many developers working with DotNetNuke will go as far as modifying the core, quality control during this migration process will be critical.

Content

Contributor Experience

With respect to usability, DNN typically does not fare well, but it is not terribly difficult to use, either. Of course, some personal preferences also come into play, but it is indicative that one customer said, “This UI gives me a migraine.” DNN attempted to address this to a certain extent in the version 6 update, and the contributor interface was somewhat improved. However, pop-ups replacing clickthroughs for tasks may not please everybody.

Using the DotNetNuke Content Localization Management feature in combination with a variety of language packs developed by the community, you can localize the UI — although not every module has been localized across every language. Just note that the quality of these language packs can be questionable at times, so test, first. Also, you can’t really use more than one interface language in a single install. On the plus side, the active community usually fixes known bugs and supports release cycles of language packs.

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

Contributing Content

On the whole, DotNetNuke is rather thin in content contribution. The platform allows you to add content to rich-text areas or simple, data-driven modules, but offers little in the way of enterprise controls for these processes. The value here comes in the breadth of modules, rather than the depth of features.

The default editorial interface supports inline editing, with module edit ability that is security trimmed to your role.

In 2010, DNN switched from using the open source FCKeditor, to OEM'ing the ASP.NET version of Telerik's "RADEditor," which offers Word-like rich text editing. It supports simple copy and paste from Word, to a certain degree helping users deal with the often messy clean-up process. Spell check and standard layout tools are provided with familiar iconography.

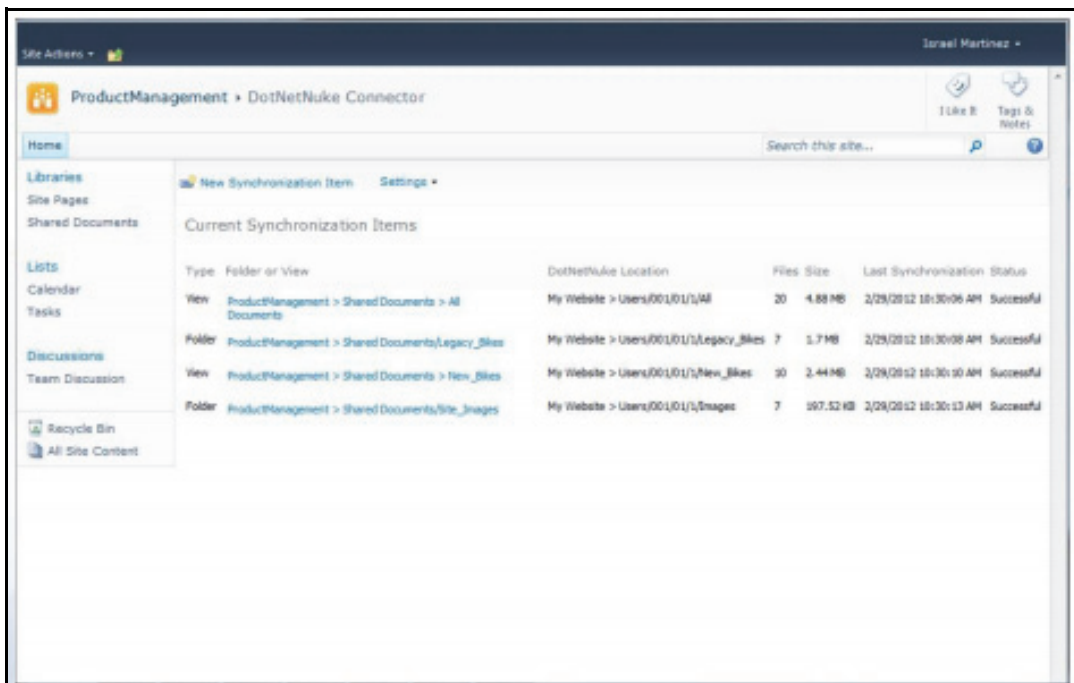


Figure 355. DNN's Enterprise edition supports scheduled import of SharePoint-managed documents.

You can import Office files using WebDAV (via an additional module), or if you are a SharePoint user and are licensed at an Enterprise level for DNN, there is a connector that supports SP versions 2007 and 2010. This connector can import documents managed in SharePoint on an ad hoc basis or against a regular schedule. You can import single files or entire folder structures as required.

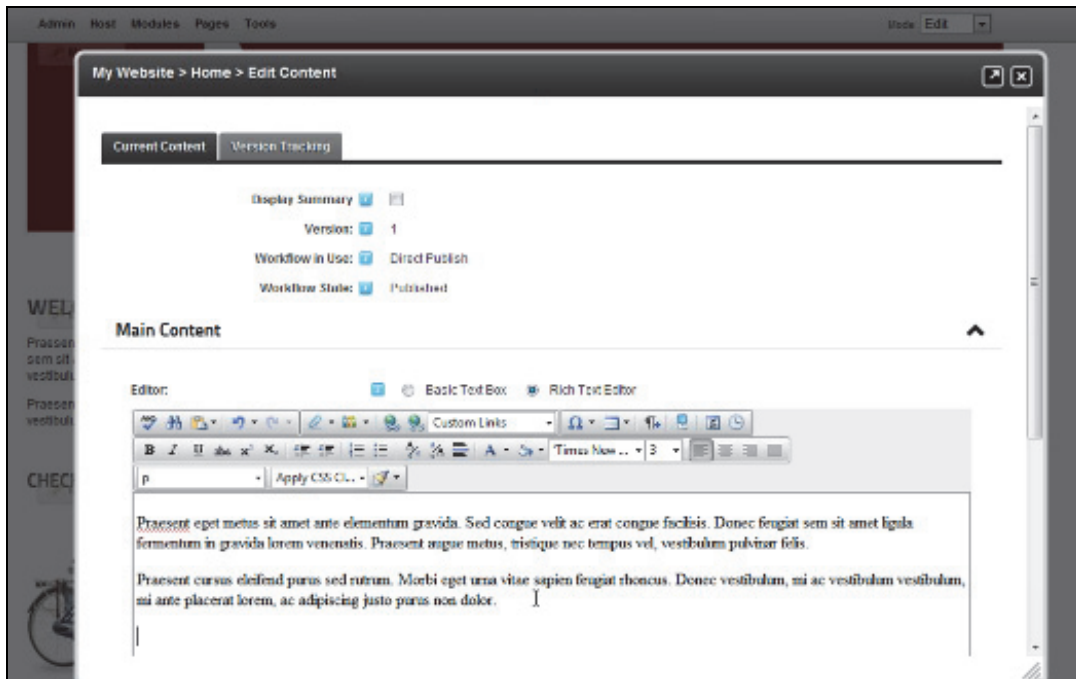


Figure 356. DNN's version 6 overhauled the content editing interface.

DNN isn't well suited for content reuse. Among other things, it lacks meaningful where-used reports. In addition, you cannot compare different versions of content side-by-side to see what changes have been made.

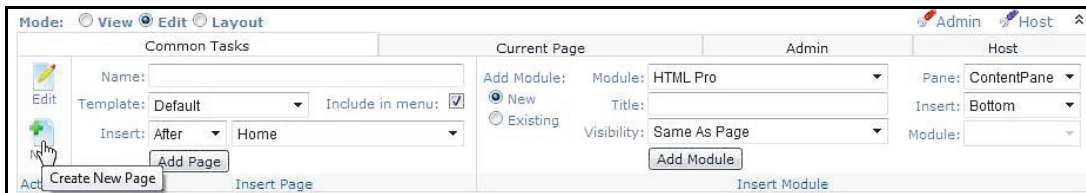


Figure 357. Authorized editors and administrators have access to a ribbon bar that allows you to add new and modify existing pages, add modules in a particular slot (“container”), and perform administrative tasks.

The default modules for images and documents are generally considered weak — this is not a great tool for intranets or highly graphical sites — but you can find some effective third-party modules here, including free ones. In those (increasingly rare) cases where you want to build a very small intranet with only scant document management and collaboration services, DNN might work, but the product's real strength lies in public-facing websites and extranets in the specific scenarios we describe in the introduction.

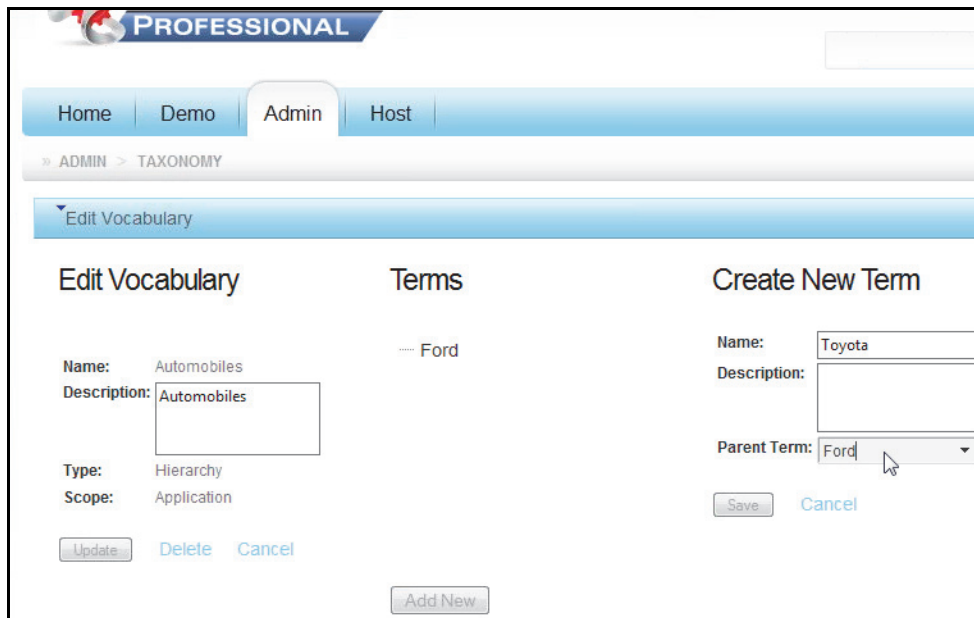


Figure 358. Limited vocabulary and tagging management is available via the Taxonomy Manager module.

Similarly, there is little support for controlled vocabularies, and indeed, metadata feels like an afterthought in this system. You can also purchase an optional module that purports to offer tagging, tag clouds, and hierarchical classifications.

Content Lifecycle

Versioning and workflow have long been absent from the software, a gap filled by third-party modules, but starting in version 5.1 released in 2009, DNN introduced both, albeit in rudimentary forms. The HTML module will now start versioning — up to a point — on each “save” action. You can view previous versions and roll them back, but the system doesn’t provide version compare.

Workflow is straightforward; HTML fields can have various “states,” and specific user groups are allowed to sign off on a stage. In its most rudimentary form, the community edition has only two stages: “Draft” and “Published.” For more advanced workflow functions you’ll have to turn to the Professional or Enterprise editions, or start custom development if you want to add a “Reject” state to send the item back to the original author. An editor saves the item, and checks the “publish” box; a reviewer can then use an “approve” button on that page to push the content live.

The fee-based “Professional Edition” allows you to add more states, assigned to specific user groups. For example, you could have “Draft,” then “Legal” approval, and finally “Published.” You can add any amount of stages, and save the edited workflows as new ones (which can be assigned to a site). However, the out-of-the-box workflow will always be strictly sequential and linear; you can “approve,” and let the content proceed to the next stage, or “reject.” Branched or conditional workflows are not achievable.

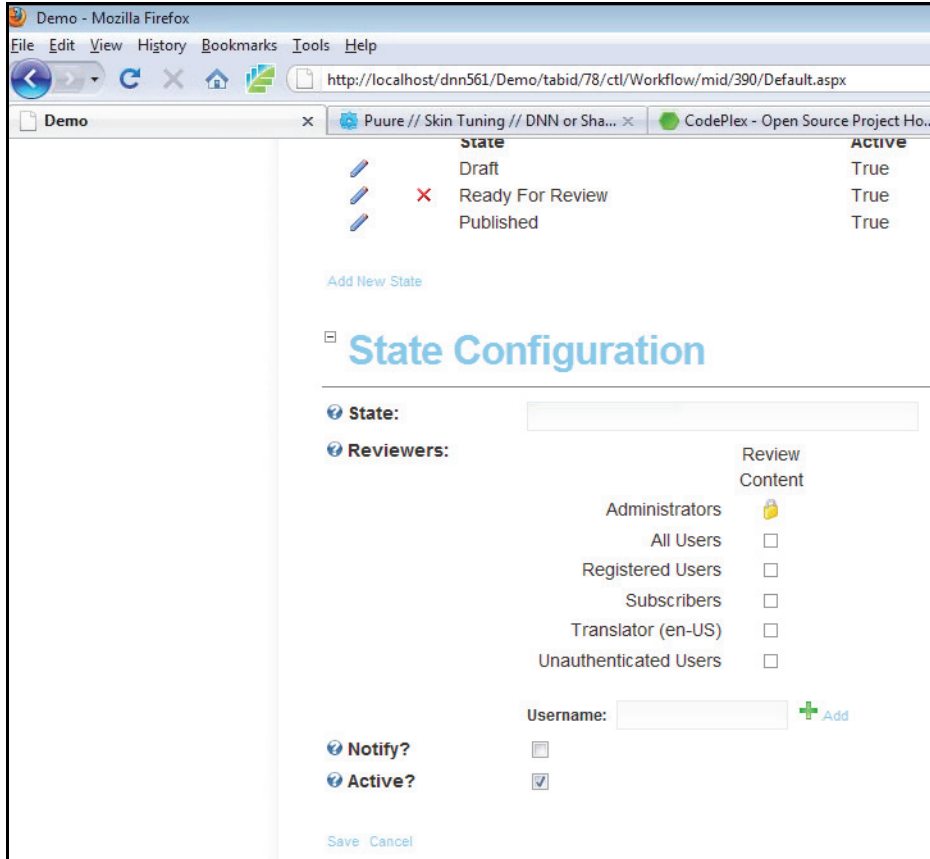


Figure 359. DNN's workflow editor allows assigning the “state” to a group of reviewers. The “Advanced Content Approval Process” in the Enterprise Edition is still a very basic workflow.

The DNN Enterprise edition (available for an additional fee beyond the Professional version) allows you to create custom workflows, with as many review points or reviewer users as necessary. On the surface, this might seem like a compelling idea, but you should carefully evaluate potential headaches that come with overly complex and custom workflows – editorially and from a development perspective.

Out of the box, DNN is not very well suited for multilingual implementations and globalization scenarios, despite some native multilingual support features and various additional modules. Even then, the system won't offer the kind of parallel workflows needed to support a smooth globalization process.

Surprisingly, DNN offers no real way to have an overview of your sites, portals, and page structures. The tree view that is common in many other products is absent. There's a module that will show all the pages in a portal, but only as a plain text list. DotNetNuke sites seem to grow organically rather than planned, and grow they will, with editors adding new pages as they're required. If you want tight governance over the structure of your sites, DNN (much like SharePoint) is probably not the right tool for you.

Experience

Publishing

With DNN, you can implement staged publishing, where content can be reviewed in-context before it's made public. However, current customers note that this process is not as straightforward as it could be.

Like most portals, DNN natively produces ugly URLs, with query strings and *default.aspx* files to load content. The typical URL produced by DNN looks like this:

```
http://www.domain.com/
Default.aspx?tabid=1&ctl=edit&mid=12&ItemID=1
```

However, you can enable a setting in the system to turn on “friendly” URL generation. Then you can output something like this:

```
http://www.domain.com/Home/tabid/1/ctl/Edit/mid/12/
ItemID/1/Default.aspx
```

If you're wondering whether that's really much better...well, so were we.

For \$100, you can purchase additional URL rewriters that will deliver meaningful, Plone-like URLs. Keep in mind though, while DNN-native URLs work with all core DotNetNuke modules, they are not always supported by third-party modules. As with any module that you're installing on top of the core, you need to do careful research before settling in any module. The modules vendor should be able to tell you whether it supports friendly URL settings.

In general, DNN won't really stand in the way of producing accessible content, but it doesn't have particular provisions to aid in accessibility, either. Much will depend on the skins you're using, or building yourself, or employing via a third-party module.

Mobile development, while possible, is neither advanced nor easy on DotNetNuke. You will still be tied to the system-native security model, and will need to create a service layer on top of the CMS.

A popular choice for those building native mobile apps on DNN is an open source toolkit called [Titanium](#), authored by DNN's neighbor in the Bay Area — Appcelerator. Titanium only works on Android and iOS at the moment. You should be prepared to spend some time setting up this project, because users report quite a few struggles in the process. Also note that Titanium is not an IDE, but simply an application packaging and testing tool.

Experience	
Publishing	
Standards Adherence	<input checked="" type="checkbox"/>
Multichannel	<input type="checkbox"/>
Mobile	<input checked="" type="checkbox"/>
Digital Marketing	
Site & Campaign Analytics	<input checked="" type="checkbox"/>
Testing & Optimization	<input type="checkbox"/>
Segmentation & Personalization	<input type="checkbox"/>
Social Media Integration	<input checked="" type="checkbox"/>
Promotional Campaigns	<input type="checkbox"/>
Community & UCG	<input checked="" type="checkbox"/>
Workplace	
Collaboration & Networking	<input checked="" type="checkbox"/>
Dashboard	<input checked="" type="checkbox"/>
Ancillary	
Site Search	<input checked="" type="checkbox"/>
Online Forms	<input checked="" type="checkbox"/>
Module Ecosystem	<input checked="" type="checkbox"/>

Digital Marketing

DNN itself doesn't offer web traffic reports, but there's functionality to plug in Google Analytics; this is particularly true for the Professional Edition, which has a more detailed level than the Community Edition. Also as part of the Google Analytics Pro package, there's some rudimentary segmentation, where you can set the rules to separate roles and pages for a more detailed analysis of your web traffic.

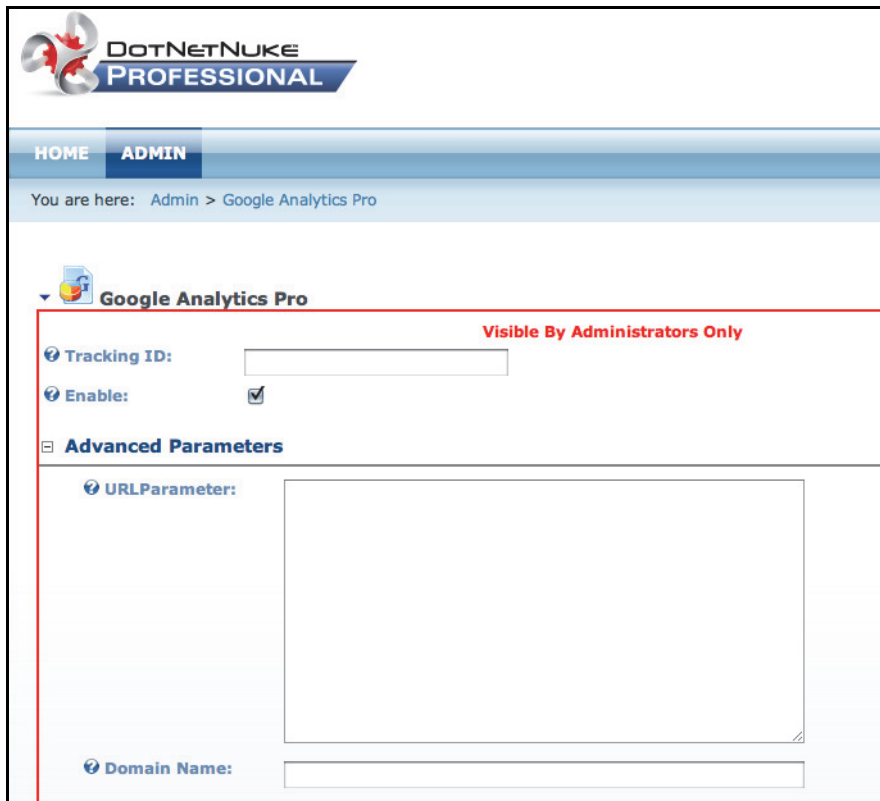


Figure 360. Even with “Advanced Parameters,” plugging in Google Analytics in DotNetNuke is not much more than pasting the tracking ID.

DNN offers some light personalization. Depending on what permissions you grant, visitors can have some control over what they see. For example, after logging in, they can maximize and minimize module views. Just as in some other portal packages, participants can manipulate portlet views.

DotNetNuke has made some strides in its latest release with the integration of Active Social elements into the core platform. Active Social was acquired in 2011 and is a long-time social and community module maker. Active Social's profiling, groups, and messaging system, combined with a “Social API,” is designed to propel DNN toward becoming a social platform with Enterprise-level functionality, albeit one primarily designed to be deployed by the mid-market tier.

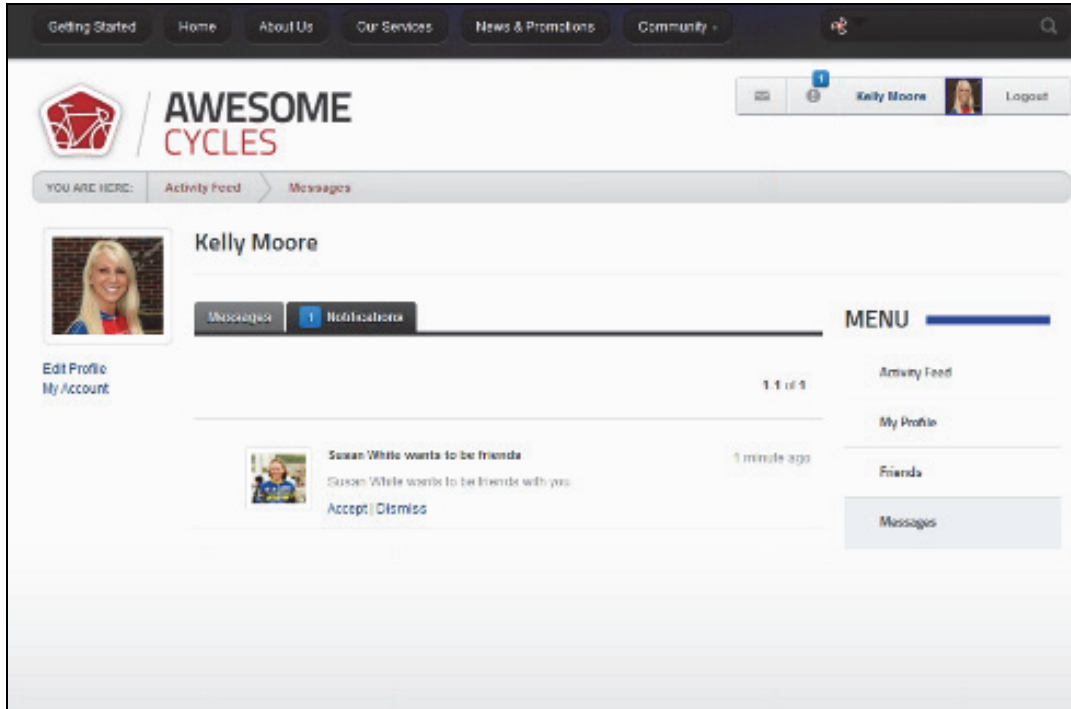


Figure 361. DNN's new social features include a messaging inbox for notifications.

Functionally, these social elements have most of the basics covered. You can have activity feeds to follow communities you have signed up for, you can follow and interact with individual users, and you can post comments to threaded discussions. Users are not tied to individual communities or sites, and can be shared across multiple sites (authenticating via OpenID if required).

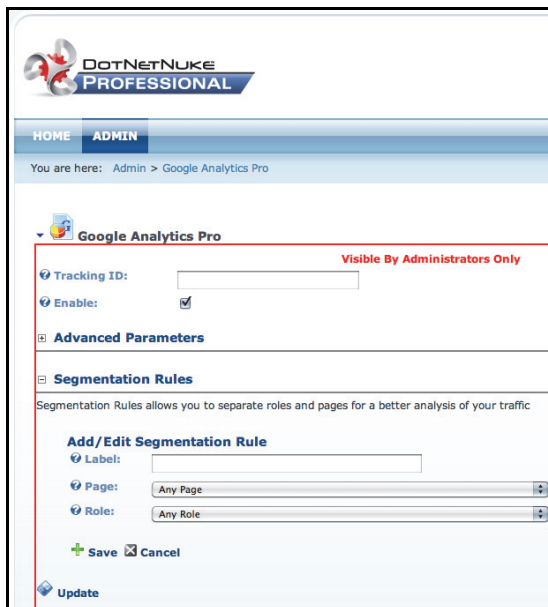


Figure 362. Segmentation rules are attached to pages and roles that are identified in the system. This is not really true segmentation compared to other systems, where you can serve up personalized content to various audience segments.

Moderation functionality is light; you cannot have message-level moderation or approval capabilities.

The “Social API” is aimed primarily at the developer community to develop extensions to these new features. However, the API is in its early days; don’t expect much from it, not the least of which is due to Active Social’s previous domination of the module market in this area.

Ancillary Services

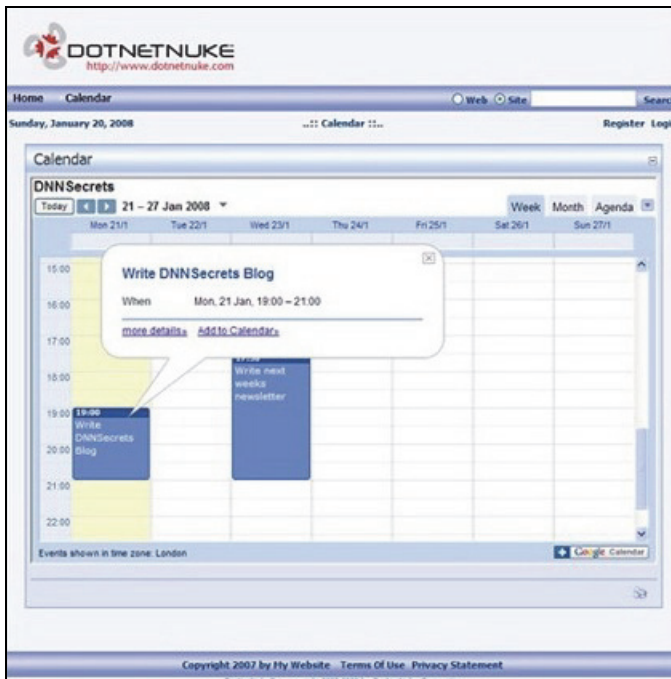


Figure 363. External services like Google Calendar can be imported via iframes.

The default search service in DNN is not well loved. It can index most modules (that’s good) and offer combined results, but seems to offer poor relevancy and performance. Relevancy shortcomings may reflect the poor metadata opportunities inherent in the package. You can license third-party, DNN-specific search tools for the canonical \$100 fee, though many licensees use Google Search (with its simple — but persistent — page spider).

DotNetNuke is distinguished by its breadth of modules, including blogs, wikis, forums, chat, and even a well-regarded storefront. However, not all DNN modules are created equally, and here

again, there tends to be an emphasis on informality rather than controls. Many customers caution that you must conduct extensive research before buying a module. For example, the survey module works nicely, but there’s no simple way for a non-technical person to clear test survey data when making it live. On the plus side, the native blog feature is helpful and employed widely. It offers moderation, CAPTCHAs, team blog facilities, and other useful goodies.

Vendor Intangibles

DNN seems most popular in North America (where there are an impressive two-dozen local user groups — though some are quite small), with Western Europe as a distant second, with some local user groups in South America. DNN exhibits a surprisingly modest profile in the Nordics, given that region’s substantial .NET footprint.

Intangibles	
Vendor Services	
Vendor Professional Services	🟡
Channel Partner Services	🟠
Support & Community	🟢
Strategy & Roadmap	🟡
Viability & Stability	🟠

The biggest difference here though, is the heavily (though not exclusively) commercial nature of the DNN community, beginning with the DotNetNuke Corporation, and extending to a large bevy of skin and module providers; some sell commercial code, and others are focused more on support. The emphasis here seems to be less on “free” and more on “low cost.”

Starting at the beginning of 2009, the DNN Corporation also offered a for-a-fee version, DNN Professional Edition (the open source version is now called Community Edition). For \$2,500 per instance, per year, you get same-day (online only) support, documentation, and a “tested and certified build of DotNetNuke.” Note that an “instance” is “An instance of DNN held in memory.” That is, if you run multiple instances of DNN Professional Edition on the same server (for instance, using virtualization), you’ll have to license these separately. Currently, this certified build is identical to the Community Edition, with the exception of some additional features like extended workflows and granular permissions. It seems that DNN Corporation is still struggling to keep functionality limited to its for-fee version.

In 2010 DNN announced the Enterprise Edition, which came with additional functionality (like the SharePoint connector and staged publishing), a range of upgraded support options, phone support, priority support ticket handling, and assistance with installation and upgrades. If these elements are critical to your organization, you must purchase the Enterprise Edition to use them.

As with other tools in this category, DotNetNuke’s source code is publicly available, under a rather generous BSD license for the Community Edition. (You are discouraged, however, from touching the core). If you want to access source code in the Professional or Enterprise Edition, you will have to buy Elite Support at \$3K/year, since those products contain proprietary module code.

Documentation is widely available, and in fact, you can pay for fairly comprehensive commercial docs. However, some developers have criticized it as being behind schedule, thin, or example driven, without suitable API references (ironically, most documentation of other systems suffers the opposite problem). Novice developers have a tendency to struggle, and like Drupal and SharePoint, you need to inculcate yourself in the particularities of the system — partly by trolling numerous blogs — to begin to get a handle on the manifold inside secrets. At the same time, DNN has inspired more than a dozen books, ranging from “Dummies” guides, to advanced developer tutorials.

Module combinations are so varied — in terms of impact, size, provenance, and integration points — that you can find numerous advice blogs on how to select and evaluate modules. The same is true for skins.

Platform upgrades come regularly, with bug fixes and minor enhancements in point releases, and security patches coming quickly in lesser dot-releases. Experienced DNN administrators don’t upgrade quickly, but instead wait to see what the community reports, particularly in connection with module compatibility.

DotNetNuke the corporation keeps a fairly tight hand on the core codebase, although the community boasts numerous superstars and gurus working elsewhere. Indeed, the community is unusually active — though not as raucous, perhaps as Joomla! and Drupal — and it is well dispersed. You will look long and hard for a large consultancy that supports DNN, but you may not need to go far to find a user group or an individual consultant.

Conclusion

In the end, DNN doesn't offer much in the way of integration with legacy systems. It does, though provide an application development platform, with an emphasis on offering a wide array of integrated applications.

In some areas like workflow and search, DNN compares unfavorably to SharePoint, but it offers a broader array of native services (e.g., Storefront, Newsletter, Banner Ads) for more of a website-in-a-box experience. However, its lack of Office integration and poor document management renders it less suited for intranet scenarios when compared to SharePoint.

Of course, larger DNN implementations employ more than its native modules. "There's a replacement module for that," you'll often hear. This is true, but as with SharePoint, third-party modules will complicate your life — and will hit your wallet, as well.

DNN's greatest asset lies in its large, active, and entrepreneurial commercial ecosystem. With earlier editions of SharePoint failing to assume the .NET mantle, DNN clearly filled an important vacuum. Of late, though, one gets the sense that SharePoint is sapping at least some energy from the project. Meanwhile, for public community sites, Drupal adds a bit more functionality, if you're comfortable living and working in the PHP world (many .NET developers are not).

In the meantime, DotNetNuke offers a potentially powerful but under-featured platform. If you go this route, make sure you keep adequate .NET development talent handy.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Joomla!: Joomla!

joomla.org

Project at a Glance

Specsheet	Joomla!: Joomla! 2.5 Summary
Geography	Global
What's New	<ul style="list-style-type: none"> • Multi-database support with the addition of MS SQL Server • Further enhancements to upgrade process • Search now based on a separate index, rather than database
Strengths	<ul style="list-style-type: none"> • Sites of low to moderate complexity can be rolled out quickly • Friendly, GPL 2 license • Can generate standards-compliant XHTML markup and human-readable, search-engine-friendly URLs • User interfaces are Spartan but friendly-feeling, easy to navigate • Supports wide range of authentication services: LDAP, OpenID, Gmail Authentication, or Joomla!'s own authentication • Modules can be reused and tailored to deliver customized content on each page • Product benefits from a large, growing, vibrant ecosystem • Many third-party extensions available • Developers with the skills needed to customize Joomla! (i.e., knowledge of PHP, SQL, XHTML, CSS) are not hard to find
Weaknesses	<ul style="list-style-type: none"> • No true workflow capability • Poor support for content reuse and re-purposing • Lack of support for custom content types make it an unlikely candidate for all but the simplest of sites • No support for multi-server environments or multisite management • Low levels of community support for SQL Server option at this time • Very tightly coupled, single-point-of-failure design makes it a poor choice for enterprise-wide deployments • Persistent security concerns and comparatively less savvy customer base renders Joomla! a magnet for possible attackers • Surprisingly uneven quality control for an open source project; frequent patching and attention required • Recent versions have had very small support windows • Community is somewhat fractious and anarchic • Joomla! users will see shorter version support windows, in a likely pyrrhic attempt to push faster upgrade behavior
Potential Fit	Simple Informational, Community Oriented, Basic Digital Marketing
Unlikely Fit	Mid-range and Complex Scenarios
Compare To	Drupal, eZ Systems, TYPO3
Operating Systems	Windows, Linux/UNIX

Repository	Database: MySQL, Microsoft SQL Server
Client	Browser: IE and Firefox, TinyMCE WYSIWYG editor for rich text editing
App Platform	LAMP
Licensing	Open Source (GPL 2)
Ownership	N/A

Summary

Joomla! is an easy-to-learn, quick-to-deploy application built using PHP and MySQL. It arose as a spinoff from another popular open source project, Mambo. Since the split, Joomla! has become the dominant sibling and boasts an extremely vibrant (albeit fractious) community.

Although Joomla! has been used to build many different kinds of websites, it remains most popular for building small websites, a role in which it excels. Generally, people use it for building personal, club, or school sites — and that’s about all we’d recommend it for it to do. Nagging concerns over security vulnerabilities — and the general enterprise-unfriendliness of the product, with regard to scalability, inflexible deployment architecture, and almost a total absence of workflow support — rule out Joomla! for any kind of serious business use, even if you have an army of PHP developers and administrators available.

To be fair, Joomla! has added more “enterprisey” features, such as an elaborate Access Control System and a new categorization scheme. However, to the extent that Joomla!’s features remain shallow, it’s still a simple product that many adherents love. Nevertheless, it’s not quite durable enough for larger organizational scenarios.

One of Joomla!’s key strengths (other than its freely available source code and liberal GPL 2 license), is the size and vibrancy of its community, and the resulting large number of third-party extensions and plugins.

With the sole exception of Drupal, you won’t find a bigger or more energy-filled ecosystem around any other open source CMS project.

Introduction

Joomla! evolved from “Mambo” (www.mambo-foundation.org), a popular open source project created by an Australian company called Miro International Pty Ltd. In 2005, owing to differences of opinion on how to govern the project, Mambo’s core developers split with Miro

Scenario Fits

Simpler Site

Informational

Microsites & Landing

Mid-Range

Basic Digital Marketing

Mobile Site

Community Oriented

Complex Site

Advanced Marketing Portal

Global Enterprise

Multichannel Publishing

Ultra-Large Single

Digital Workplace

SMB/Departmental

Global/Enterprise

International and created a separate project, ultimately named Joomla! (which means “all together” in Swahili).

A completely re-factored codebase and improved API emerged with version 1.5, but concerns about security vulnerabilities continued to dog the product. To be fair, this is partly due to vulnerabilities found in third-party extensions rather than Joomla!’s core codebase, and open-source software projects tend to be more transparent about security issues than COTS offerings. However, it still raises concerns. We suspect Joomla! could be targeted for attack because it tends to be downloaded by novice administrators.

The Joomla! community has achieved greater velocity than its older sibling has. In fact, it’s precisely the size and energy level of the Joomla! community that makes it such a compelling alternative to Mambo, and such a vibrant alternative to more “controlled,” commercial open source patterns, like eZ.

Version 1.7 came out in July 2011, just six months after the previous dot-release, as the Joomla! community adopted a schedule-based release plan, rather than its previous ready-when-it’s-done, feature-based releases. As a result of the short release cycle, 1.7 did not bring in any substantial new features for the end user. Architecturally, however, this release marked the formal segregation of the base layer — the former “framework” now known as Joomla! Platform — from the higher-level Content Management System (CMS) layer.

The community now makes a distinction between long-term and short-term releases. A short-term release of the Joomla! CMS is slated every six months and supported up to the availability of the next release; every third release will constitute a “long-term release” – supported until three months after the next long-term release. The underlying Joomla! Platform will differ from this schedule, and is slated follow a 3-month release cycle.

The latest version 2.5 was originally slated to be version 1.8, but given the large number of fixes and improvements (over 300), it rolled from a minor to major version release. The Joomla! team now says that “long term releases” will be released every 18 months.

The result of this change is that versions 1.6 and 1.7 are already out of support, with the largely used (but now elderly) version 1.5 falling out of support in April 2012 (security fixes will continue until September 2012). While this will force the community toward less fragmentation, there will be a great deal of pain within the community to reach version 2.5 — especially for those making the journey from 1.5 — dependent as Joomla! users are on the extensions added to ensure the continuity of key functionality.

Technology

Technical Administration and Security

Joomla! in many ways typifies the simplicity of the LAMP experience. Not all open-source systems are easy to install and configure, but Joomla! is. Even non-technical users find it a breeze to set up, and Joomla! won't force you to look for programmers to help create a basic site.

For anything more than a basic site, of course, you'll need to get into the code, but even if you need to extend the system with third-party modules and custom code, you'll be up and running much faster with Joomla! than with almost any other alternative.

Fundamentally, Joomla! is a set of PHP scripts that run on a regular web server (e.g., Apache), backed by (typically, but no longer exclusively) a MySQL database. All of the text-based content that it manages is stored in the database, whereas scripts, images, style sheets, and miscellaneous system artifacts are stored on the web server's file system. Additionally, in v2.5, a search index was added, replacing the database queries that previously powered searching both for back-office and web front-end purposes.

Joomla! 2.5 is based on multiple, logical (not physical) tiers, consisting of a Platform (formerly Framework) tier, a CMS (formerly Application) tier with three sub-classes (JSite, JAdministrator, and JInstallation), and an Extension tier. The sub-classes of the Application tier correspond to three main applications — Front-end, Back-end, and Installation.

Historically, the framework and app tiers were not completely independent entities; however, from release 1.7 onwards, the two tiers were formally segregated:

- **Platform** – Represents the core services required by any application, such as security (authentication and authorization), database interaction, file system handling, logging, and exception handling
- **CMS** – Represents the higher-level application tier that exposes traditional Joomla! functionality

Note that both tiers will be developed independently of each other while following their own release cycle and naming conventions. This means that the administrators would have to handle more frequent updates with Joomla! installations; independent updates to Joomla! Platform and CMS could also have potential conflicts, resulting in additional work for administrators.

This separation was the most important change in release 1.7 and a significant one too; you can now choose to use traditional Joomla! as is, or develop custom web applications over the Joomla! Platform, without getting tied down by the restriction of the CMS layer. This could be useful for boutique website development firms, or designers looking to move beyond the

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

product’s limitations. Moreover, a standardization of interfaces between the two layers will offer a welcome change for the Extension developers.

On the downside, you should know that if you build custom apps on top of the core Platform, you are forking yourself from the CMS trunk and losing the advantage of being a part of that large community. Moreover, it also means that you now have to track two separate development streams (with respect to updates, bugs, stability, performance, etc.), and there could be unwelcome dependency problems with modules that rely on both tiers. To that extent, this once “simple” product just got a bit more complex to administer.

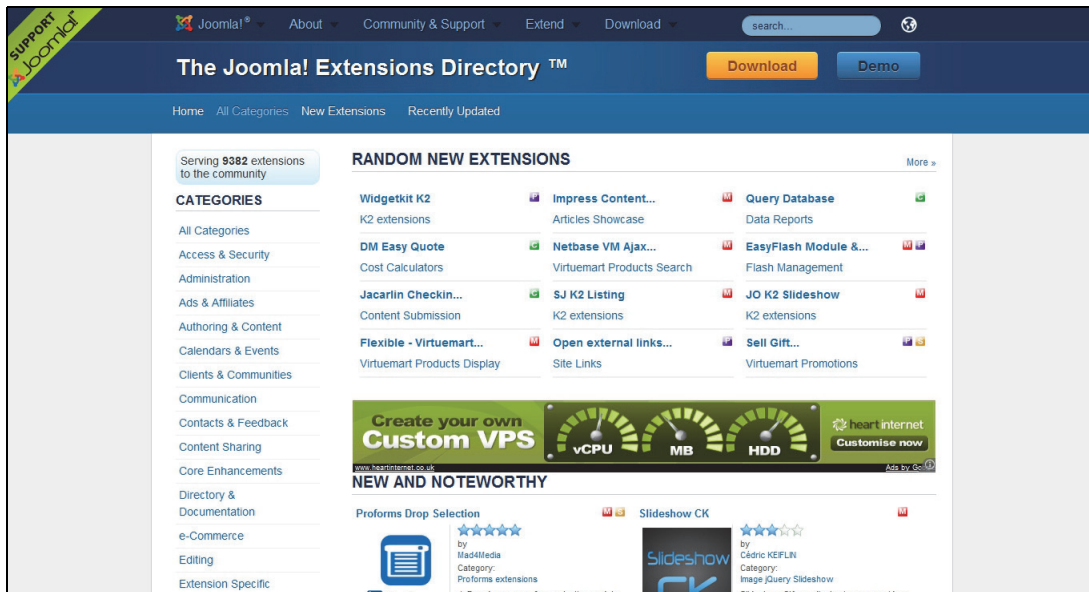


Figure 364. Joomla offers over 9,000 extensions available to the platform as of April 2012.

The architecture is a strongly coupled one, in the sense that you use the same instance of Joomla! for content management and content delivery. In other words, your site visitors and content entry people all use the same Joomla! instance. Conversely, a single installation of Joomla! can be used for only one site; there is no notion of multiple sites or multisite management. This lack of architectural flexibility will (by itself) rule out Joomla! for many potential customers.

Because of the tightly coupled architecture, Joomla! delivers content dynamically from the instance that manages the content. Based on a request, a page is dynamically generated and displayed. Joomla! can also render all content to PDF (and print-friendly formats) out of the box. However, you should test this process with the actual pages you intend to deliver, to see if it’s reliable for your particular needs.

Joomla! publishes PHP and HTML pages via the Apache web server. At runtime, the PHP interpreter assembles the final HTML from a combination of static files and resources pulled from the database. Depending on how dynamic and complex your web pages are, there might be 3, 10, 20, or even more queries into the database for each page served. This can create performance problems in a hurry (see below).

Note that because of the very tightly coupled architecture, Joomla! has no concept of staging, per se. Once an item is live, authors make changes to the content directly on the live environment. This is great for real-time previewing, but terrible for formalized publishing processes and security. In theory, you can create an environment with multiple Joomla! installations in a staging-production setup, using replication and third-party extensions. However, it is painful to create links or interlink content items in such a way that they work in this kind of environment, and it still lacks dependency reporting or referential integrity checks. Again, consider carefully how much “coupling” you’re willing to tolerate and what it might take to overcome the challenges posed by too tight a coupling between management, production, and delivery.

Joomla! has a concept of “front-end” and “back-end” users and their related access permissions. Version 1.6 introduced a completely new Access Control and Security system, allowing you to have any number of groups and access levels — and you could assign users to multiple groups, with variable inheritance.

The good news here is that the ACL and Security system provides a very fine-grained access control mechanism, suitable for enterprise scenarios that demand a more sophisticated security setup. However, with flexibility comes complexity, and you will need to invest time to plan how to structure these controls.

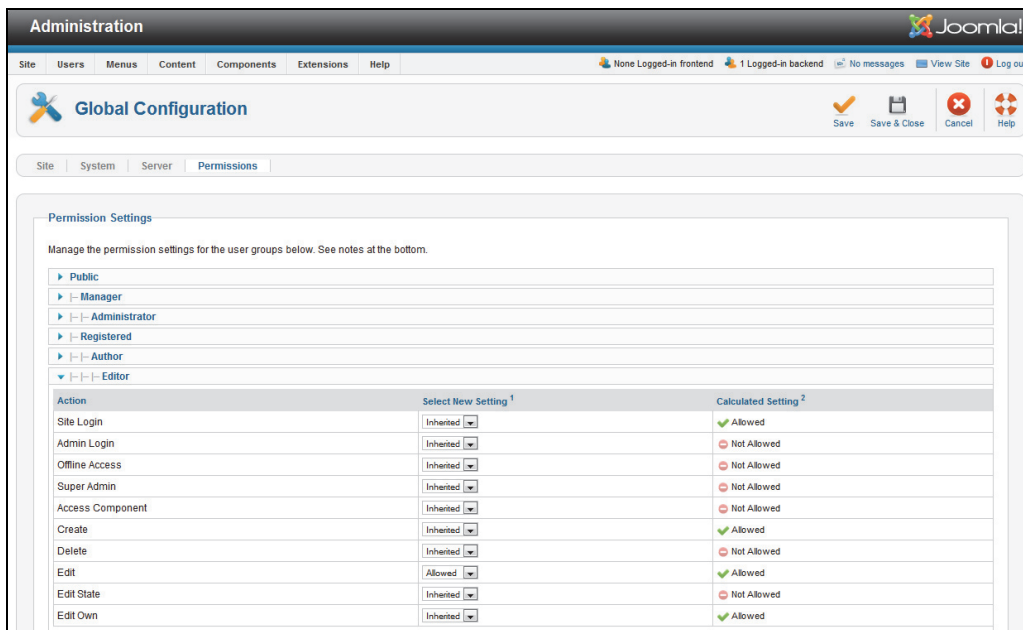


Figure 365. Granular ACL support introduced in version 1.6.

Joomla! supports a variety of authentication methods (LDAP, Gmail, OpenID, and native Joomla! authentication), although the native authentication module is the default (the others require a bit of work to enable).

System reporting is one of Joomla!’s most widely lamented functionality holes. There are few reporting extensions for Joomla! and no obvious standouts. Most users run queries directly against the MySQL database, but this is not onboard reporting.

There have been persistent reports that Joomla! is prone to security attacks. To an extent, these can be attributed to the fact that there is no governance to audit and certify the extensions that are developed by the open community, which can lead to potential security risks. Joomla! has attempted to mitigate these risks to a degree in recent releases.

- Users are now forced to move to newer releases due to the predefined support window for each release (six months for the short-term release, and twenty-one months for the long-term release). This will ensure that they have updated security fixes.
- There is now a one-click upgrade process for Joomla! CMS and another for extensions. It also supports core database updates (like the addition of new tables and changing a field), thereby making it easier for admins to upgrade to the next release.
- With the abstraction of the Joomla! Platform & CMS layer, it'll be easier for developers to add new features to the Platform and CMS. Adding a separate search index (i.e., searches are not performed against the database) reduces the ways in which injection attacks can be attempted against the platform. Whilst this does not remove the possibility of attacks, it does significantly reduce the risk.
- CAPTCHA (a system for filtering robotic login attempts) has now been added to the core CMS application, rather than an optional extension that users had to add manually. This adds slightly to the basic security credentials of a vanilla Joomla! installation.

Development

You use a UI called Template Manager to apply a look and feel to your website (Figure 366). Each template can have one or more styles. A style is essentially a variation of a template, which enables you to create a different look for your pages. A very nice feature is that you can achieve a completely different look and feel by assigning a style to an individual page or a set of pages. This is a useful feature when you want to display different parts of the site with a different look and feel. The template code and associated CSS files can be edited directly from the Template Manager web interface, even though they reside on the file system rather than the database. This is helpful, especially in a hosted environment, where you do not want to download and upload files repeatedly for small changes.

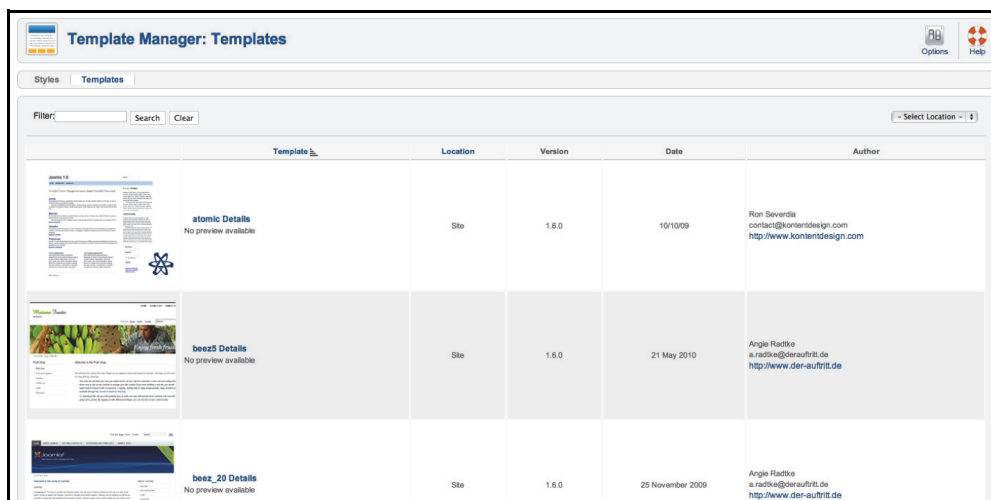


Figure 366. The Template Manager UI allows you to manage templates and styles.

Much of Joomla!’s power is derived from add-ons called “Extensions.” There are five main types of Extensions: Languages, Components, Modules, Templates, and Plugins. Plugins are more advanced extensions — essentially event handlers. In the execution of any part of Joomla!, whether it’s the core, a module, or a component, an event is triggered, and any plugins that are registered with the application to handle that event will be executed.

Since the product itself is completely open-sourced under a liberal LGPL v2 license, you can alter Joomla!’s internals anytime. The code is well documented online.

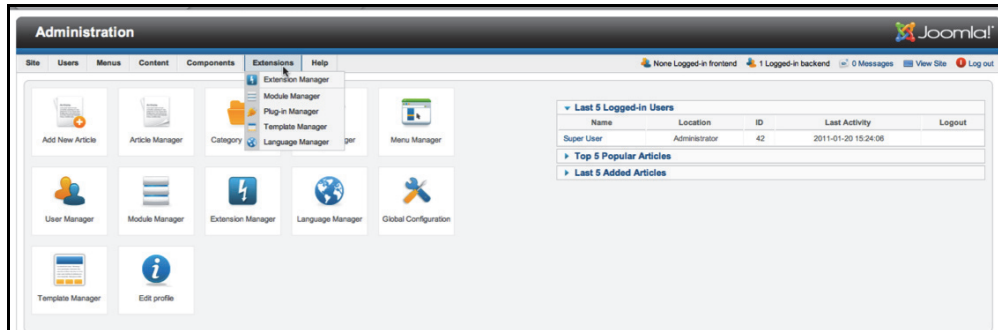


Figure 367. You can manage all Extension types from a single web interface.

Templates in Joomla! are a multi-part affair and must be created in a separate text editor, such as Adobe Dreamweaver. A template is usually an XML file, a set of PHP files, and all associated assets (CSS and image files). These files have to follow a particular file and directory layout and can’t be installed by hand into the file system — unless you also know how to make the requisite database entries. Most people package a template folder tree as a .zip file and let the Template Manager do the actual importing. This can become cumbersome after a while, but the procedure is not overly difficult, and the template pieces tend to be short and easy to read, with *template.css* often the most complicated piece.

A new forms API enables developers to customize the forms that make up the back-end. Thus, you can have your own custom fields in forms such as the content entry form. While this in no way comes close to the ability to define custom content types (such as those in Drupal using CCK), it does make it easy for an extension developer to create an extension that will customize your content entry screens.

Performance

While Joomla! is by no means universally slow, performance issues are quite common. The issues tend to be database related: too many queries per page, poorly formed queries, badly structured database tables, databases that must be rebuilt, and more.

The system’s repository (wherein all page content is stored in the database, rather than in files), in conjunction with a module-driven architecture (in which every module can make multiple SQL queries) provides many opportunities for performance degradation. It is a tightly coupled system, which doesn’t help.

Historically, Joomla! only allowed one option for its repository database: MySQL. With version 2.5 however, multi-database support was available for the first time, with Microsoft’s

SQL Server being the first alternative option. This has the added appeal, in that it allows Joomla! to be deployed for the first time to Microsoft’s “Azure” Cloud offering.

Joomla! claims to be close to releasing drivers that will additionally support PostgreSQL, Oracle and PDO (PHP Data Objects). PostgreSQL seems to be the closest to formal support, but it is important for any of the non-LAMP elements, since their use is likely to be scarce within the community as a whole, making peer support more of a challenge.

Security problems have dogged Joomla! for most of its existence, and many of the actions required to harden the system against attack also exact a performance penalty. Whether or not the more enterprise-friendly SQL Server alternative will improve matters here is unclear — but worth considering — especially if in-house capabilities exist on that platform.

Unfortunately, scalability is not a strong point, so don’t count on solving your transactions-per-second problems by scaling out, unless you can run multiple instances and keep them all in sync. You’ll need to become a MySQL expert or a brave SQL Server pioneer at the same time.

Content

Contributor Experience

Usability is widely deemed better for Joomla! than it is for many well-known, open source WCM offerings. Recent releases have a particularly friendly look and feel that seems to appeal to many users. Some nice, subtle improvements enhance efficiency. For example, in almost all of the screens there’s a new option called “Save and New.” This is very useful when you are creating multiple items (content or users) because in just one click, you can save an existing form and open a new form to enter the next set of values.

However, be aware that the system’s real-world usability will vary quite a bit depending on your use-case scenarios and the types of add-on modules you’ll be using. Community-built extensions differ considerably in usability. Carefully test the extensions you intend to use before deciding whether Joomla! meets your usability requirements.

Joomla! has adopted “Mootools” (a popular JavaScript framework) for an improved look and feel and snappier performance, utilizing v1.4 within the most recent version of the CMS. Most screens have been redesigned to use fewer clicks, and the out-of-the-box cosmetic appearance has been refined. The product initially may *feel* usable, but play with it a while (and test your real-world use cases) before you make any final judgments.

The worldwide Joomla! community has been active in creating UI localization packs — literally scores of them. Joomla! has decent support for Unicode (specifically UTF-8), which means better OOTB support for internationalization, and better support for right-to-left (RTL) languages. (Remember: that’s only for the core product; third-party extensions may or may not support UTF-8.) The installer can install Joomla! automatically in many languages.

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

Alternatively, you can download the language pack of your choice, and add that to the Joomla! back-end, using extensions. This enables you to maintain the back-end in multiple languages, thus enabling users to have the back-end in their chosen language.

While it is possible to create a front-end that is WCAG and Section 508 compliant, the back-end (administration) is still not compliant with these accessibility requirements. Joomla! requires an extensive re-write to its core for it to be completely compliant and currently plans to do it in the 2.x release cycle. For now, a start has been made with a new “Hathor” template for the administration back-end, in version 1.6, and is more accessible than previous releases. Remember that Joomla! is fundamentally a “composite app,” consisting of modules and extensions from a diverse community of contributors — some of whom are less concerned than others about accessibility, Unicode compliance, and so forth. This is one of the prices you pay for an open source product that is not firmly controlled by a commercial sponsor.

Contributing Content

Joomla! provides an administrative application for creating and managing content, along with other production tasks. However, content can also be created, edited, and published directly from the front-end, without logging into the administration back-end. The content entry screen (Figure 369) consists of text fields, WYSIWYG controls, and fields for metadata entry. You can add access-level information, publishing specifications, links to menus, and other parameters that define what displays when you see the detailed article. You can even view statistics, like how many hits that specific content item has received.

Note that out of the box, these fields are fixed; you can’t add your own fields. In fact, you can’t create your own custom content types. If you need this feature, you will have to depend on third-party extensions or write your own (assuming that you have developers that know PHP and understand Joomla!’s extension framework). Fortunately, the Joomla! online forums host active discussions about this.

You create a page by assembling a list of available extensions (components, plugins and modules) and combining them with “Menus.” A Menu defines the navigation structure. Using the Menu Manager, you can create Menus and assign content items to them. This is a manual process, and new items that must be linked also must be added manually.

HTML content can be entered using either a plain text editor, TinyMCE, or the CodeMirror editor. TinyMCE (a pure JavaScript-based WYSIWYG editor with no ActiveX or Java-applet dependencies) is preferred by most customers.

Because there’s no spell checking with these editors, users must integrate another extension for this (although many browsers now come with an integrated spell checker that works on any text field). This is where a more powerful, applet-based editor can be helpful, but that’s an integration project you would have to undertake yourself.

Joomla! allows for nested categories. You can have a category tree — as deep as you need it to be — but the old limitation that one article can only belong to one category still remains. While nested categories are a good addition, the fact that there are no sections means the URLs must undergo a change. This may have implications on your SEO initiatives, as well as general content migration issues. Overall, the system rates poorly in content reuse.

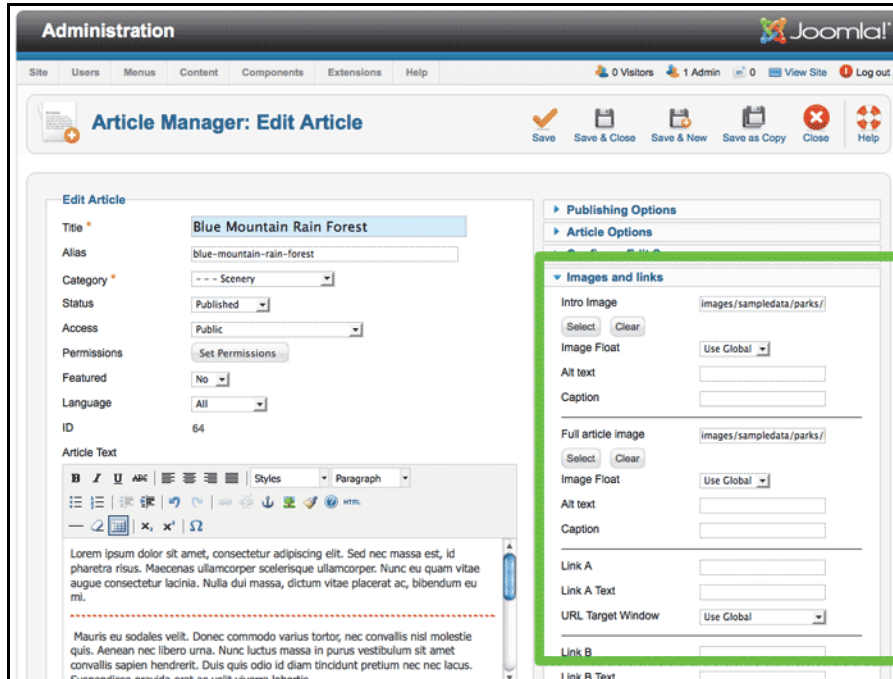


Figure 369. Joomla!'s “Article Manager” lets you specify images and links within article text.

Joomla! has standard check-in and check-out features, but also a “Global check-in” privilege, where an administrator can check in all the checked-out files. However, you must exercise caution while using Global check-in, because there is a chance that some authors might lose their changes.

Note that Joomla! has a very simple and elegant way to define where a particular module displays. Each template defines positions such as left, top, right and so on, where a module can appear, and each module can be assigned a position.

Although no one would accuse Joomla! of being a Digital Asset Management system, Joomla! does have a very useful Media Manager for uploading images and other media files (Figure 370). You have the ability to select multiple files and upload them using a Flash uploader (a feature that had caused issues in legacy releases and had to be removed). You can also bulk-delete multiple media files, which was not possible in earlier releases.

For repository search, Joomla! comes with six search plugins, corresponding to Categories, Content, Contacts, Newsfeeds, Sections, and Weblinks. However, there's no way to perform a custom search based on metadata. This is the kind of thing you have to implement for yourself using PHP and SQL. The docs.joomla.org website has a tutorial for this.

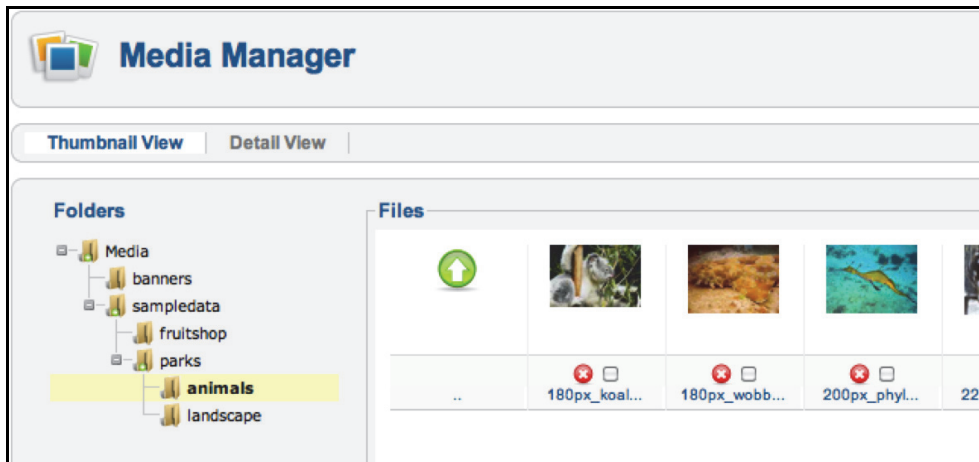


Figure 370. The Media Manager enables you to upload or delete multiple files.

Content Lifecycle

There's a simple Create → Edit → Publish workflow, in which an author creates content, an editor checks it, and a publisher publishes it. However, if you need anything more complex (e.g., graphics or legal as another step in the workflow), there's no easy way out, short of diving deep into the code and writing your own solution. The built-in “approval” facility is just that: It's a simple approval feature, rather than an actual workflow. There's no task list, for example, and consequently, there's no way for editors to know if stories are pending approval.

Content can be filtered according to different parameters, such as category, author, language, and access level.

Joomla! is not designed as a multisite management tool and would not be a good choice for globalization scenarios. There is a commercial product, Jentla (sold by Australian firm ZacWare), which adds multisite capability to Joomla!, along with additional scalability options. Jentla, however, hasn't yet built the kind of track record that suggests real-world suitability for large-scale deployments.

There is no “retention” in Joomla! per se. Content can be deleted (trashed) *or* “archived”—an important distinction in scenarios where there are articles that are important, but not current. There is no automated removal of expired files, however.

Experience

For search-engine-friendly URLs, Joomla! offers three options (and a check-box UI in the Global Configurations dialog for setting them). The first option collapses Joomla! URLs (which can be hundreds of characters long) to something fairly short and readable, but containing a PHP filename and .php suffix. The second option (which relies on Apache's *mod_rewrite*) removes the PHP filename. The third option adds “.html” to the end of URLs to make them more acceptable to search engines.

Multichannel

Because a single Joomla! installation can only be used for one site, it is not possible to reuse the same content for multiple channels (intranet, Internet, mobile, and others), unless you're adept at developing PHP apps.

Segmentation & Personalization

The Access Control system can help you model your site in such a way that you only show specific content to specific audiences (groups). Other than that, personalization facilities are absent from Joomla! out of the box, although various members of the community have posted information about their experiences in implementing various types of personalization, and some one-off implementations have proven the feasibility of using Joomla! underneath an interactive, portal-like front-end.

Site & Campaign Analytics

Joomla! provides some basic site analytics. Statistics such as most popular items, number of hits, browsers, operating systems, domains, page impressions, and search texts can be logged and tracked. It is also possible to integrate with other statistics packages like AWStats, using third-party extensions.

Site Search

New in v2.5 is a “Smart Search,” a plugin search, such that Joomla! users do not have to rely on applying extensions or exposing the repository for SQL queries via PHP. As such, this search supports proper stemming, so using wildcard characters (something that had been eradicated from virtually every other search tool a decade ago) is no longer required. Whilst not enabled by default (a plugin is currently required), apparently to maintain backwards compatibility, it is designed to utilize the CMS' event system for maintaining state within the index.

Experience	
Publishing	
Standards Adherence	<input type="checkbox"/>
Multichannel	<input type="checkbox"/>
Mobile	<input type="checkbox"/>
Digital Marketing	
Site & Campaign Analytics	<input type="checkbox"/>
Testing & Optimization	<input type="checkbox"/>
Segmentation & Personalization	<input type="checkbox"/>
Social Media Integration	<input type="checkbox"/>
Promotional Campaigns	<input type="checkbox"/>
Community & UCG	<input type="checkbox"/>
Workplace	
Collaboration & Networking	<input type="checkbox"/>
Dashboard	<input type="checkbox"/>
Ancillary	
Site Search	<input type="checkbox"/>
Online Forms	<input type="checkbox"/>
Module Ecosystem	<input type="checkbox"/>

Additionally, the plugin supports result filtering, meaning that basic faceted result displays could be implemented with additional programming efforts. However, in this first iteration of the feature, Joomla! suggests that scaling might be an issue and even advises consideration of other searches — such as Solr — for larger sites. Precise metrics of what might constitute the breaking point for “Smart Search” are not available, and users are advised to consider load testing this function before deploying it. There is a range of extensions developed and maintained by the wider community that can also provide alternative solutions.

Community & UGC

In many ways, community-oriented and UGC micro-applications represent the strength of the tool. Joomla! has built-in applications for newsfeeds, banners, contacts, polls, and web links. However, these components are sufficient to build only the simplest of sites; for anything more complex, you’ll depend on third-party extensions. You can find components for e-Commerce, banner management, collaboration, alternative search tools, and others, but the quality varies and you’ll need to test extensions for usability, performance, scalability, and overall quality.

Project Intangibles

Early leaders of the Joomla! community formed an organization called “Open Source Matters, Inc.” (OSM) to provide organizational, legal, and financial support to the project. Currently, OSM’s funding comes from a mixture of donations and revenues from online advertising. OSM allocates funds to allow core team members to attend events such as LinuxWorld’s community FOSS days and Joomla! Day.

Intangibles	
Vendor Services	
Vendor Professional Services	<input type="radio"/>
Channel Partner Services	<input type="radio"/>
Support & Community	<input type="radio"/>
Strategy & Roadmap	<input type="radio"/>
Viability & Stability	<input type="radio"/>

Although OSM is registered as a not-for-profit organization in the United States, Joomla! itself is remarkably popular globally, with the core team spread across 11 countries. The popularity of Joomla! can be measured by community participation; there are more than 160,000 members on the forums, more than a million forum posts, more than 1,000 third-party developer projects and several hundred-thousand Joomla! downloads each month.

Indeed, Joomla!’s greatest strength as a platform is its large and helpful community — certainly one of the largest of any open source project. The support forums are quite helpful, and very active. There are plenty of independent developers that will help you. Generally, you can get immediate answers — even to the most esoteric of questions — free of charge. If there is a hiccup here, it is the dramatically short support windows that the community will follow going forward. Specifically, no more patches and other enhancements to older versions once new versions are out only a scant month. This rather draconian policy was designed to spur more comprehensive upgrading — and doubtless it will — but it could also lead to confusion and certain vulnerability among those who don’t religiously upgrade.

In terms of documentation, the product natively incorporates a large amount of contextual help (and even some *extremely* verbose tool tips), plus there is a huge abundance of documentation

— in the form of PDFs and wiki pages — on the Joomla! project websites. In fact, the project suffers from an embarrassment of riches when it comes to online resources, because the sheer volume of it makes it hard to find what you’re looking for, even when you have a specific need.

As previously mentioned, third-party extensions abound for many features. Although this is often considered a big advantage, there’s a downside as well. When you upgrade Joomla!, you have to ensure that all installed, third-party extensions are compatible with the new version. Additionally, the third-party-developed extensions should be free from security vulnerabilities and have no problems with Section 508 compliance and Unicode compatibility. (A corporation using Joomla! might also worry about possible third-party license and patent infringements.)

While product upgrades and extension upgrades should become easier, do not assume this, and be sure to test before making changes to your production environment. Hence, we would caution you to avoid “early adopter” status on any major new pieces of functionality (such as new ACLs). Perhaps more importantly, with greater power could come greater complexity, and the simple Joomla! that thousands know and love could present a very different set of challenges going forward.

Finally, you should understand that — perhaps as a result of its somewhat tumultuous birth — the Joomla! community is comparatively quite fractious and decentralized. This helps the breadth of the platform, but there have been rumors of another possible fork. Even though a new leadership has taken over OSM and things seem to be improving, you should observe cautiously before investing in Joomla!.

Conclusion

Joomla! has become one of the most well-known, LAMP-based site management tools, exuding a certain charm of its own, despite more than a few shortcomings. Many will find that the benefits of PHP, MySQL, a stone-simple architecture, and a rich array of available extensions (combined with a huge user and developer ecosystem) more than outweigh the system’s simple-minded approach to workflow and poor support for custom content types — and reusable content, in general.

Because of its tightly coupled architecture and built-in propensity for performance problems (which can be overcome — at a cost), Joomla! still remains best suited for small sites. If you want to build a lightweight site quickly and are comfortable with the security, usability, and performance (as well as the caveats associated with third-party extensions), consider Joomla!.

Consider Joomla! if you want a rapid proof-of-concept prototype before undergoing a serious implementation with a heavier-weight system. It’s also worth considering if you want to start creating basic types of web content, so that by the time you decide on a production platform, your content is already in an open database, with the proper metadata.

On the other hand, if you have more sophisticated content management requirements — involving multiple sites, many authors, diverse content, real workflow, real scalability, hot failover, proper staging and QA, in-depth logging and auditing, dependency reporting, air-tight security, 24/7 support from the people who wrote the code, and so on — Joomla! is not the platform for you. Don’t try to make a Swiss Army knife do the job of a chain saw.

Ultimately, it costs nothing (but your time) to download Joomla!; give it a try. We recommend that you do just that — particularly if you're looking at a non-commercial, open source system to power a simple operation.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Kentico: Kentico CMS

kentico.com

Product at a Glance

Specsheet	Kentico: Kentico CMS 7.0 Summary
Geography	Global partner network, HQ in the Czech Republic
What's New	<ul style="list-style-type: none"> • E-Marketing edition adds a plethora of new functionality • Cloud version of the product now available • Visual workflow designer and more formalized staging services included in some versions
Strengths	<ul style="list-style-type: none"> • Unusually feature-rich product, with a plethora of modules at a relatively modest price point • Relatively easy to customize and tweak for simpler needs • Comparatively good support and documentation — a rarity at this tier • E-marketing features might be comprehensive enough for some small enterprises • Adheres to modern .NET standards
Weaknesses	<ul style="list-style-type: none"> • Breadth rather than depth of features means some functionality not fully baked when released (e.g., mobile preview) • Crowded UI may prove overly complex for casual contributors • Repository and audit services still lag in some areas, reducing enterprise value • Lack of integration among e-marketing functions reduces long-run value of those add-on modules • Bolted-on staging approach likely won't work for larger, more configuration-management-conscious customers • Overly aggressive sales and marketing tactics may prove off-putting • Not well suited for more than very basic globalization scenarios
Potential Fit	Community-Oriented Site, Informational Site, Basic Marketing
Unlikely Fit	Global/Enterprise Digital Workplace, Multichannel Publishing, Ultra-Large Single Site, SMB/Departmental Digital Workplace
Compare To	Ektron, EPiServer, Telerik, Sitecore, DotNetNuke
Operating Systems	Microsoft Windows
Repository	Database: MS SQL
Client	Browser: IE, Firefox, and Safari — Chrome not fully supported
App Platform	.NET
Licensing	Based on functional options, for one website base license at \$1,999, to one server ultimate license for unlimited sites at \$4,499, full e-marketing options at \$14,999 (or \$999p/m for the cloud version)
Ownership	Privately held

Summary

Czech-based Kentico’s eponymous CMS is a .NET, page-based system, which runs on a Microsoft architecture (Windows, MS SQL Server, and Visual Studio for developers). Its architecture resembles that of Ektron, EPiServer, DotNetNuke, and Telerik Sitefinity. Kentico remains a much lower-cost alternative to the first two, but is more substantial (and a bit pricier) than Telerik.

Where the system really differentiates itself is the wide array of features that come out of the box. None of them are particularly sophisticated, so don’t expect Kentico CMS to be of much use in the complex parallel workflows required for large-scale globalization efforts. It is also poorly suited to metadata-driven architectures with placeless content and intensive reuse. Finally, it is not well architected for larger enterprise scales.

However, for many simpler scenarios (especially community-oriented sites and simple e-commerce sites), the product may be able to cater to most of your needs.

On the whole, this is a much more capable system than you’d think at first glance — although you may have to dig a bit deeper to find out how things are done. For now, Kentico has managed to keep everything coherent; however, with its latest releases the vendor is definitely risking greater feature sprawl, and you’ll want to weigh that against your own internal capacities.

Since its inception in 2004, Kentico has been steadily carving a niche for itself, with what the company says are some 2,500 customers, running over 15,000 websites around the world. Hopefully with this growth, the company can continue to live up to its reputation for quick and helpful support and comprehensive documentation.

Introduction

Kentico Software is based in Brno, Czech Republic. It was founded in 2004 by Petr Palas (now the CEO of the company), who came from Moravia Worldwide, a large Microsoft-centric translation/localization services company. From the initial 4 employees, it has grown to its current 94: 60 employees are based in Brno, one (business development director) is in the UK, and 8 employees are in the U.S. (president of sales and operations, “product evangelist,” two sales managers, and two support engineers). The company says it has some 1,200+ partners in more than 80 countries around the world. Since this makes for an average of two licensed customers per partner, some are obviously more experienced than others.

Like Ektron and EPiServer, Kentico CMS is predominately a page-based .NET system (it has some of the facilities for componentized reuse as well, but this means digging into code). Organizations like Gibson and LiveStrong are Kentico customers, but you will also find this

Scenario Fits	
Simpler Site	
Informational	<input checked="" type="radio"/>
Microsites & Landing	<input checked="" type="radio"/>
Mid-Range	
Basic Digital Marketing	<input checked="" type="radio"/>
Mobile Site	<input checked="" type="radio"/>
Community Oriented	<input type="radio"/>
Complex Site	
Advanced Marketing Portal	<input checked="" type="radio"/>
Global Enterprise	<input type="radio"/>
Multichannel Publishing	<input type="radio"/>
Ultra-Large Single	<input type="radio"/>
Digital Workplace	
SMB/Departmental	<input checked="" type="radio"/>
Global/Enterprise	<input checked="" type="radio"/>

CMS is used by small mom-and-pop shops, as well as by agencies — a current area of focus for Kentico — and others that white-label and/or host the platform.

Because this system was designed in 2004 — rendering it a good deal younger than most of its competitors — it has the advantage of being built on a .NET platform from the start, at a point in time when .NET was growing up; the change to .NET 2.0 in 2006 was the most major interruption (and with it came Kentico CMS 2.0, only six months later). As a consequence, the company hasn't had to rebuild the core workings of the system since then, and most of its development went in breadth of features, while the architecture remains relatively modern. Because of this, it comes with a plethora of modules out of the box (including blogs, wikis, e-commerce, and most recently, e-marketing).

Technology

Technical Administration and Security

Kentico CMS is an ASP.NET system that uses MS SQL Server as the repository (binaries, such as images, can also be stored in the filesystem for performance reasons). The system doesn't require the latest versions of the Microsoft infrastructure — one of the few .NET systems in this report that will actually still run on Windows XP — using the built-in IIS5, MS SQL Express 2005/2008 as database, and Visual Web Developer / Visual Studio 2008 for development. Of course, in practice, you'll likely want to pick a more up-to-date environment, but the low bar of requirements set by Kentico should mean it's relatively easy to outsource your hosting — or simply set up a laptop development environment.

Kentico CMS publishes dynamically, rendering using either its original ASPX templating (which had just been added to .NET when Kentico developed CMS version 1.0), or the “portal engine,” introduced in version 2.0. Note that this is not a true portal, akin to we cover in our Enterprise Portals research; it's not a strict container with portlets. Rather, the “portal engine” uses Web Parts, which can be added and configured in a visual design mode.

Similar to DotNetNuke, one of the major drawbacks traditionally in using Kentico CMS for larger, enterprise-y deployments was its lack of good support for the separation of live/production and staging/test servers. Recent versions have addressed this shortcoming to a certain extent, but having separate environments still feels far from a core part of the product, and it requires a considerable amount of planning and manual intervention. By default, the staging module is not deployed and has to be added manually — post installation.

In reality, most customers still seem to install the package in one instance on one box, which means that changes are made to the same repository without the ability to test inside the firewall first. This results in a much higher probability that untested code or unapproved content will end up on the public-facing, live server. This difficulty in decoupling a tightly coupled environment efficiently may not sit well with many IT departments.

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

In theory, there's a staging module that is supposed to help separate content editing from the live server, utilizing a manual or scheduled sync between the content staging server and the live server(s). Beyond that, there's no ability to publish to staging. You always have to sync (the staging environment in reality is a sync target, no different to any other development environment); the publish function is the first step before you can do any syncing. Until then, the staging module will not recognize that data.

It is clear from Kentico's own documentation that this process requires a great deal of planning and technical effort to achieve. If you need to decouple Kentico CMS into a more traditional Development/Testing/Acceptance-QA/ Production (DTAP) setup, this is an area where you should carefully test the functionality — in detail — before committing to the product.

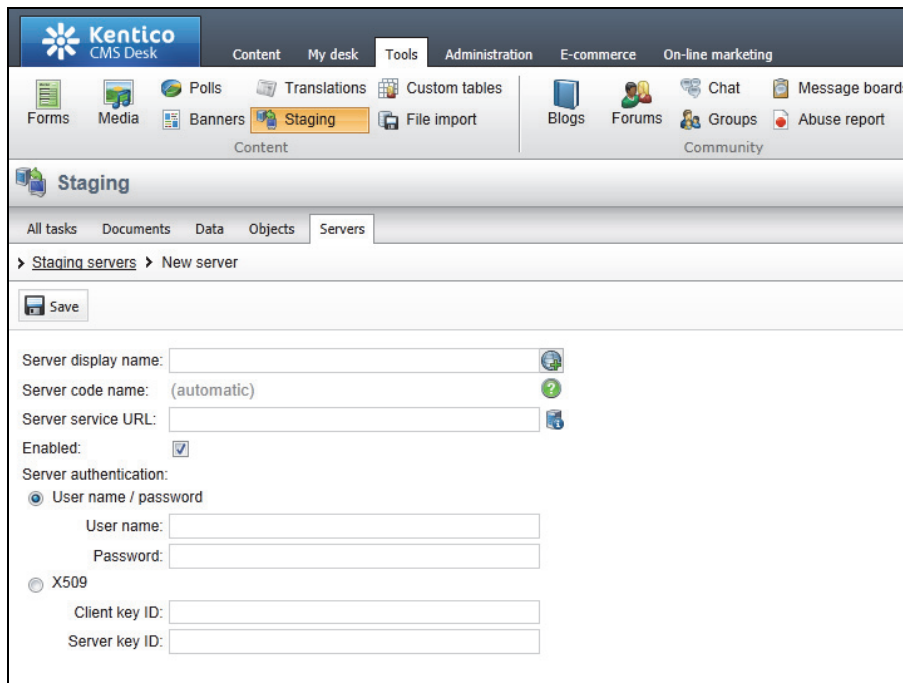


Figure 371. A lot of manual work is required to administer Content Staging in Kentico CMS.

In late 2010, Kentico announced support for Windows Azure cloud computing platform and has since added support for Amazon S3, where it offers its own SaaS offering of its separate e-marketing “EMS” product. The same code base and installation routines get used for all deployments, whether hosted on-premise, remotely, or via Kentico's supported offering.

For authentication, the system can use its integrated user database (which also has an ASP.NET user provider), Windows authentication, or LDAP. The vendor also offers an Active Directory import utility (which maps groups to roles in Kentico CMS), though, of course, you'd then have to schedule updates, as well.

Note that the previously described optional staging functionality does not support the synchronization of account data, so authentication for your staging server must be handled as a separate importation task. Not a total deal-breaker, but a bit of a kludge all the same, particularly since the same user directory will also be used for social features (such as blogs or

wikis). This probably explains why social content (such as blog posts) are also not supported by the staging synchronization service.

Users are global (i.e., they will exist in all sites in a Kentico implementation). They're assigned to roles, which have specific permissions for one site. The system contains sample roles, and the example "Corporate Site" comes with 12 predefined roles. You can also add your own or modify the existing ones.

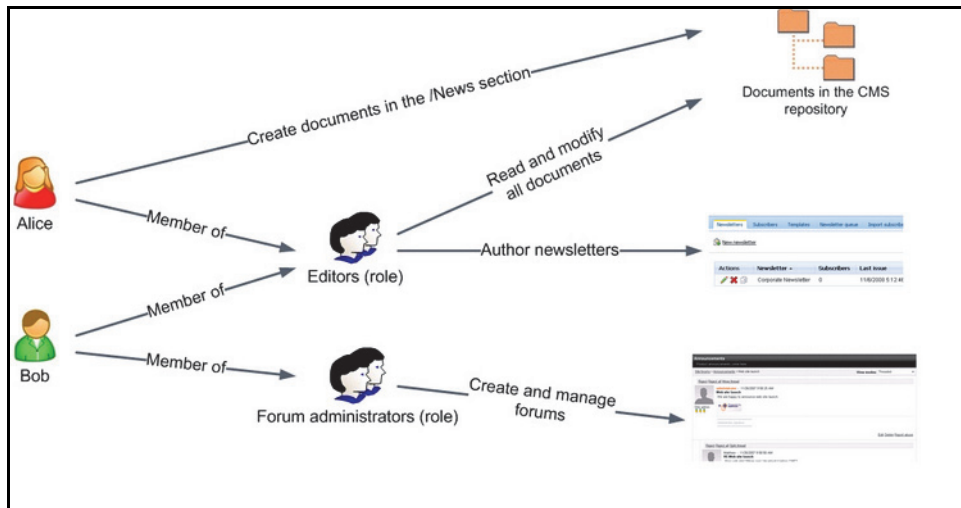


Figure 372. Kentico roles and permissions can be complex, and affect visitors' rights. Source: Kentico.

Permissions (such as "manage," or "read") can be set on modules, document types, or "document-level." For example, you could give "modify" rights to an "editor" role, for the document type "news." This would mean that all editors would be able to modify any document of the type "news," wherever it resides in the individual site. However, you could also set the "modify" permission on the "news" section in the site tree, which would mean that "editors" would have the permission on any document of any type in that section of the website. These "document-level" permissions can be inherited — which means that they are valid for a top-level item and any children — but they also can be set at an individual document level — a relatively rare capability.

All of this means that Kentico CMS is very flexible in setting permissions in your implementation. However, the combined effects can become increasingly hard to fathom since users may have various roles, roles have both document type and document-level rights, and to top it off, user-specific check marks get into play such as "is administrator." In general, rights add up rather than cancel out. For example, if a user only has "modify," but not "delete" rights on an item, but another role grants him "delete" rights on that specific document, he will have both rights. In short, if you don't watch out, users will be allowed more than might've thought possible.

Other Windows-centric systems allow similarly complex schemes; for instance, Sitecore offers a special visual tool to show the effects of the combined access rights in the content tree. However, Kentico currently doesn't have such visualization, and you'll have to check each item's properties to find out. Plan carefully in advance, rather than having a casual administrator toy with these properties.

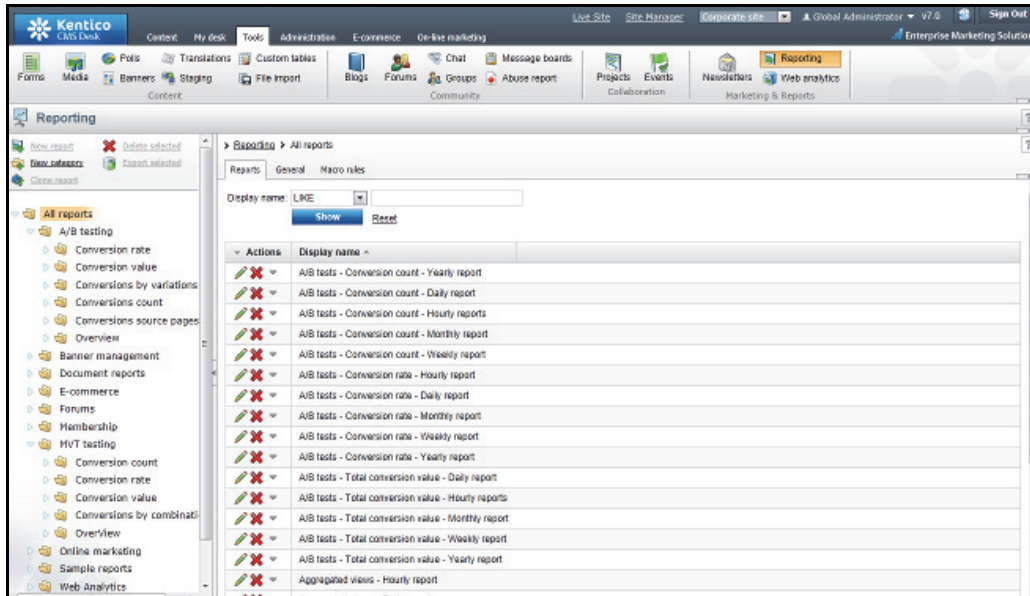


Figure 373. Kentico provides a plethora of reports out of the box.

The system has extensive reporting; it comes with dozens of predefined reports, both on internal activity (such as “checked-out documents”) as well as external and visitor behavior (such as “Number of sales today”) for the e-commerce module. The addition of the e-marketing EMS modules adds further reporting for the marketing automation and A/B testing/segmentation tools.

Additionally, it’s relatively easy to create your own reports, though this usually requires some code to extract the information from the database. Of course, these are usually straightforward lists of items that match the query, but many competitors in this tier only offer logfiles, which you have to parse yourself.

Development

As mentioned before, developers have the option to use either the ASPX templating module (which requires Visual Studio for development), or the newer “portals engine,” which allows less technically savvy users to define a page through a tab in the editorial UI. For complete control, you could still go with standard ASPX development, but Kentico recommends the portals engine, since you can add in your own code quite easily.

With the portals engine, you first create a layout page. Users can then add (configurable) “Web Parts” to the page from the interface (somewhat similar to SharePoint). As with any widget-like page builders, developing the Web Parts takes more development up-front, but if you manage to create them as multi-purpose, you could put page building in the hands of the power users, rather than requiring a developer for each new page layout. The advantage of Kentico is that it already comes with many Web Parts, and we’ve been able to find quite a few for free online, as well.

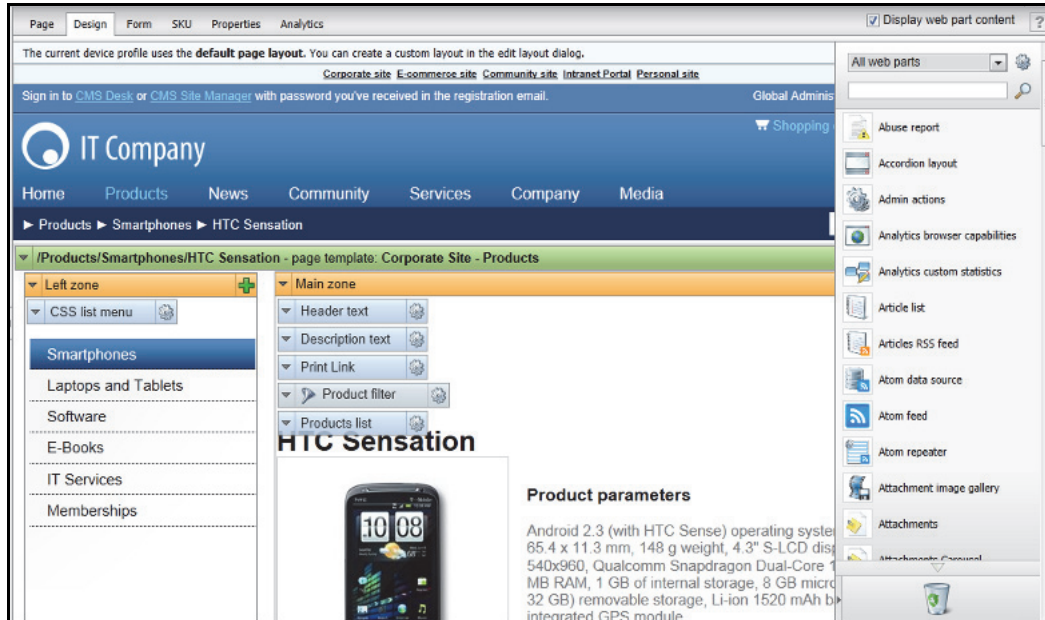


Figure 374. Designing a page using Web Parts with Kentico’s portal engine.

You create content types using a wizard, which walks you through the process. In the background, this will actually create a new database table populated with the fields you’ve defined. You can also select a custom icon for a document type; this is unusual for this tier, and it is a useful tool for editors working with content later.

The corresponding editorial UI is built from the document type definition, so for each content field you must define both the database field type (i.e., integer or text), the default value, and editor attributes (i.e., radio button, default values). Field validation can be done using regular expressions. As with much of Kentico CMS, the wizard isn’t quite as slick as it first appears (for instance, you can’t really go back one step if you forget something), but it allows you to do most of the things you’d want to do with a run-of-the-mill site.

For each document type, you will have to create one (or more) transformations to display the content in various pages. It’s possible to use XSLT for this, but most implementations seem to use ASPX controls — which .NET developers should be much more comfortable with using. Fortunately, the wizard also will create a simple, straightforward transformation, which makes it easy for novice developers to modify it, rather than creating a new one from scratch.

Some traditional ASP.NET developers might be disappointed to find out that it’s difficult to extract a web application layer out of your Kentico CMS project if you want to exercise a more straightforward MVC (Model View Controller) model in your development and separate business logic from presentation and other application layers. This means that for now, you will have to stick to the ASP.NET Web Forms pattern for creating web applications in lieu of the ASP.NET MVC framework.

Multisite management is quite good. Using the CMS Site Manager, you can create new sites and manage their properties. Sites are treated as separate entries, e.g., they have their own roles, while users exist in all sites; workflows or document types are defined globally. This

seems to match customers’ scenarios well, and several companies host dozens of (small) sites from the same Kentico instance.

Performance

The system’s performance, on both the back-end, and front-end visitor facing website is described by most customers as “fine,” and we got the same sense running it on a virtual machine in tests. It’s perhaps not the leanest solution ever, but in most cases it’s not really problematic, either.

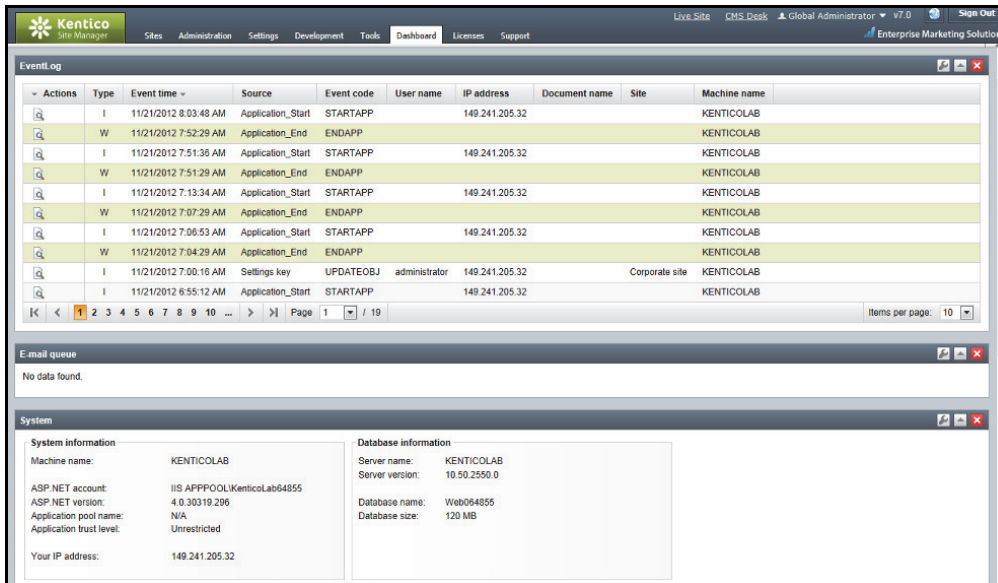


Figure 375. Kentico Site Manager offers a decent view on current system events.

Most customers seem to adopt “standard” optimization techniques used in any Windows/IIS/MS SQL/.NET environment. This often involves a lot of tweaking of IIS processes, and maybe scaling out the database to another machine and/or clustering it, but none of this is particular to Kentico CMS.

To be fair, sites built on Kentico CMS aren’t usually particularly complex or traffic intensive, compared to some of the other products in this report, such as Escenic and CoreMedia, which specialize in global, high-demand scenarios. To scale to the demands of an international news site would probably be taxing; then again, this isn’t the scenario Kentico tries to address.

One minor annoyance is that the ASP.NET pages are compiled on the fly — i.e., they get compiled at first request, which is relatively slow, but subsequent requests will be much quicker. As a “frying” system that integrates front-end and back-end in the same core, the same goes for the back-end interface; this means that some (lesser-used) pages in the interface will load noticeably more slowly than others will.

In practice, this should only affect small implementations with few users and an under-dimensioned server, but it is a hint that you should still do your own testing, especially if you plan to run the CMS on your own existing server. Likewise, don’t expect all of Kentico’s modules to perform equally well out of the box, since their functional specifications will have a very different impact on server performance. The e-commerce module, for instance, has

been described as fine for a smallish web store, but not up to the task of running a full-blown e-commerce operation.

Content

Contributor Experience

Kentico’s editorial interface, “CMS Desk,” is a relatively smooth UI, which most users seem to understand without intensive training. It contains a dashboard, “My Desk,” with direct access to a user’s work items. These include categories such as “recent documents,” “checked out by me,” and, usefully, “outdated documents,” a list of all documents older than a specified date. Oddly however, this is the second tab of CMS Desk — the interface actually opens on the more classic “Content” tab, as if Kentico didn’t quite see users adapting the dashboard *en masse*. The full interface can be a bit overwhelming, considering the massive amount of functionality Kentico has crammed in there. However, it’s possible to personalize the interface to specific roles (e.g., a casual contributor only sees what he needs to see.) You’d be well advised to invest some time in trimming the features for all but the administrator.

Content	
Contributor Experience	
Overall Usability	◐
UI Accessibility	◐
Contributing Content	
Authoring & Transformation	◐
Tagging & Taxonomy	◐
Content Reuse	◐
Media & Document Management	◐
Repository Services	◐
Content Lifecycle	
Workflow	◐
Globalization	○
Archiving & Compliance	◐

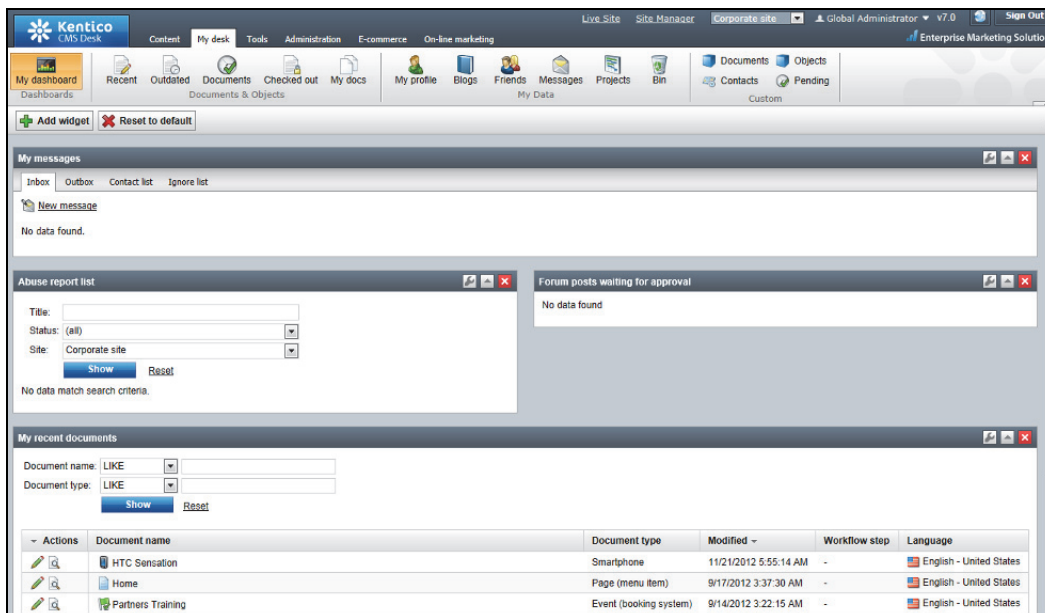


Figure 376. The “My Desk” view in Kentico CMS acts as a dashboard of events and documents that are personalized to a specific user.

The layout is a familiar one — buttons for various actions on top, the site tree on the left, and the content area to the right. The buttons (like in Sitecore) match the look and feel of the ribbons introduced in Office 2007, though Kentico hasn't gone all-out on copying the exact look as Sitecore has (which is arguably a good thing).

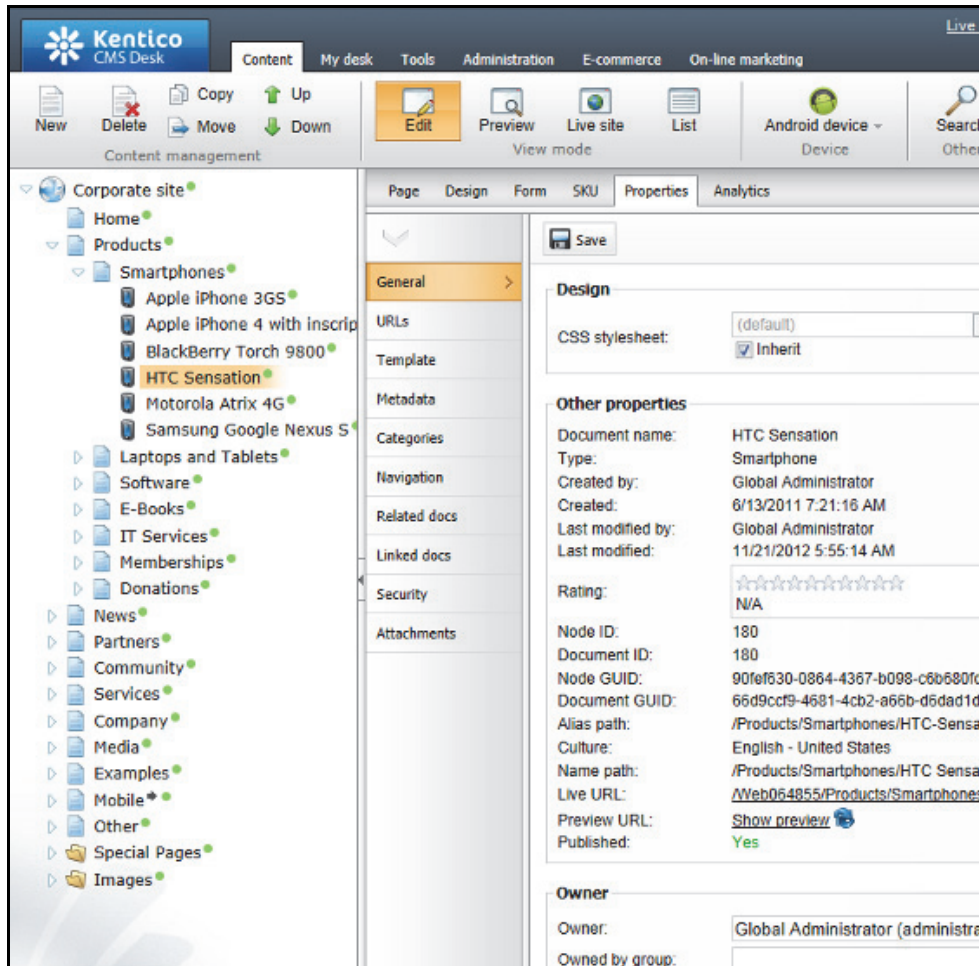


Figure 377. Kentico's editorial interface, CMS Desk, can become quite busy; fortunately, you can trim unnecessary features for specific user roles.

The operation of the content area, however, is somewhat problematic. There are some minor glitches; the “back” button of the browser has unpredictable results, and unless you explicitly set the option for confirmations, it's quite easy to accidentally navigate away from your edits and lose unsaved work. More importantly, when you start editing a page, a new set of tabs appears horizontally (the full set would be “Page, Design, Form, Master Page, Properties”) which aren't entirely self-evident. When you select the “properties” tab, another set of vertical tabs will appear, as well. This is especially annoying since this is where the “metadata” form is hidden; by that time, a user has navigated several tabs and buttons to get there. Thankfully, a user can click small buttons to hide the top buttons or side content tree and have more room for the work at hand.

Power contributors certainly should be able to grasp this in a relatively short period, but more by a process of “rote learning” than by intuition. Careful attention should also be paid to browser support; while Kentico officially supports Google’s Chrome browser from version 12 onward, our testing revealed that on contemporary versions of that browser, the interface could become somewhat unstable. Some digging into error codes revealed that Microsoft’s “Silverlight” was the likely culprit, suggesting that the UI is (to a greater or lesser extent) dependent upon that plugin being installed and functioning fully. Given Silverlight’s less than universal support, test extensively — particularly if your organization has rigid rules around browser support.

We also suspect that the interface won’t score well in accessibility, if only because of an extensive use of frames. This may be a chicken-egg matter, since Kentico seems to have few customers in government where this would have been crucial; none of the users we spoke to had even considered UI accessibility. Localization, however, is excellent, with the interface available in twelve language variants (or “UI Cultures” in Kentico speak) — including English, French, German, Italian, Dutch, Greek (a rare language to have available in an administrative interface), and, of course, Czech and Slovak. However, in some languages (depending on screen settings), the button labels will actually overlap, which doesn’t help legibility.

Contributing Content

Contributors will usually work in the “Page” tab of the content area, which displays a preview of the page. As with Magnolia, this has the advantage that users are more or less working “in-site.” The button bar for the WYSIWYG editor (Kentico uses FCKeditor) is displayed above the page, so it doesn’t distort the fidelity of the preview. However, as with other in-line text editors, this has the disadvantage of focusing the user on the presentation at hand, which can be detrimental to understanding the effects on other presentations.

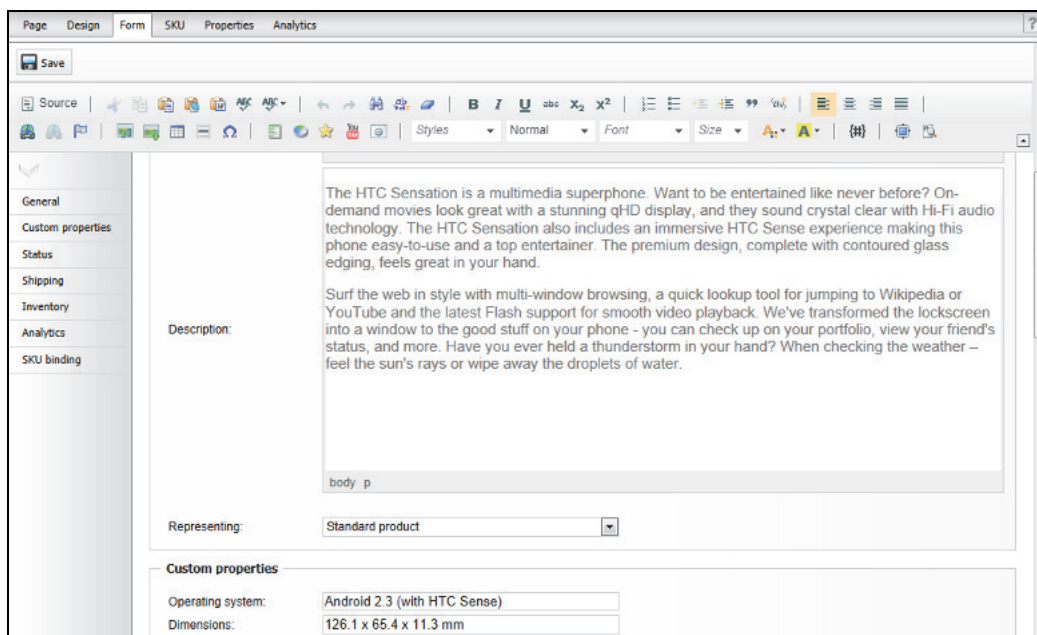


Figure 378. Editing a page in Kentico CMS, using the well-known FCKeditor.

It's possible to use Kentico CMS in a less page-oriented manner, and reuse documents across a site, for instance, displaying “news” in different presentations, such as a list and a single item page. However, the interface doesn't quite lend itself to this, as users will be editing in-line without working on a page, and content will always have to be attached somewhere in the site tree, first.

The functionality really was designed with specific scenarios in mind, such as product databases, and it doesn't lend itself well to reuse on a more granular level than a document. This isn't helped by the fact that the product isn't very strong in metadata management, and tagging with “categories” was clearly designed with simple blogs in mind; there is one flat list of categories per site with the ability to add “My categories” on the page level.

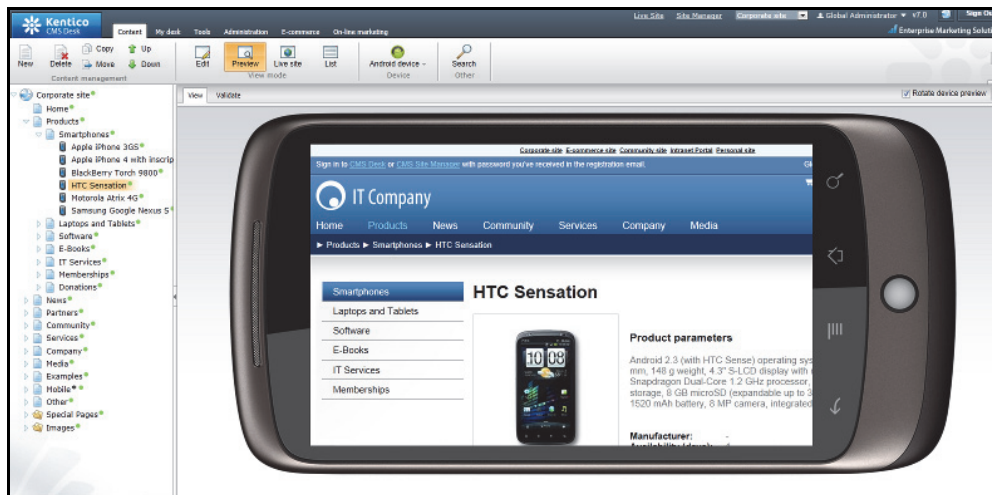


Figure 379. The mobile device preview in MyDesk is basic.

You can preview content using some basic mobile device templates — such as generic “iPad” or “Android Phone” layouts — which purport to indicate how well or poorly individual pages will look. The result of this approach is way behind what you might expect from such an emulator and is arguably misleading because of the lack of contemporary devices and little guidance as to how these results are being rendered (e.g., what is the assumptive resolution on which this device profile is based?). This is one of those occasions where no functionality might be preferable to an overly simple and inaccurate one.

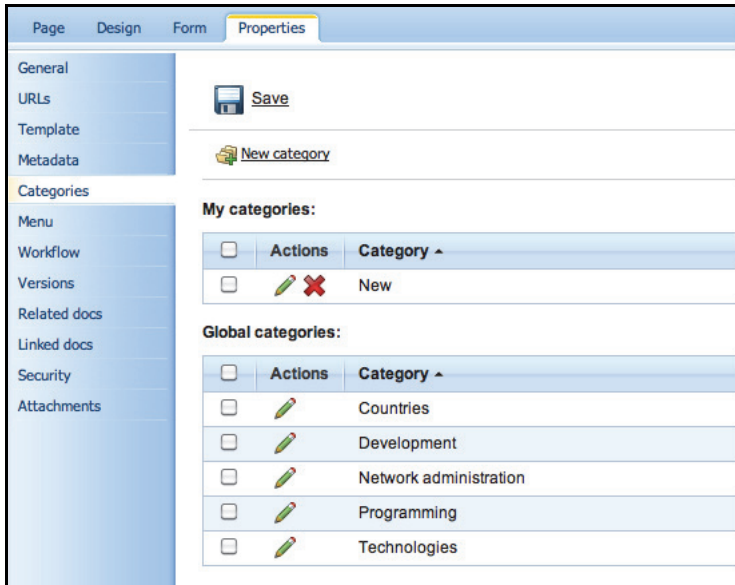


Figure 380. If your needs involve sophisticated tagging, controlled vocabularies, and categorization functions, Kentico CMS may not fit the bill with its rather flat approach.

You can add images and other media (like video) to a specific document, by picking one from the content library, or by browsing “media libraries.” It’s possible to resize the run-time rendition of the images from the properties fields. Again, as with most of Kentico CMS, it’s nice to have all of these options, but having all of them available to the user might be confusing.

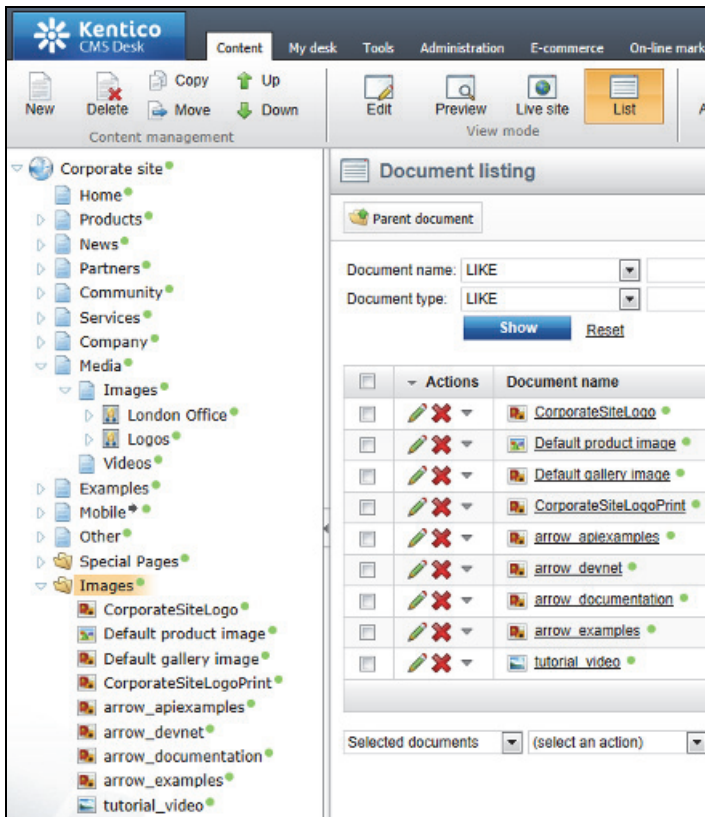


Figure 381. Working with images in Kentico CMS' Media Library Folders.

Repository search is very basic, based on SQL queries with keyword search only. There are no options to filter on metadata (other than the “only published” state). For more complex searches, you will have to resort to reports instead, which is inconvenient.

A module formally called, “Document Management Package,” adds support for simple document-management tasks, WebDAV integration for Microsoft Office applications, and some project management functionalities, and is now a default installation for the majority of product editions. For intranet scenarios (as well as competing with SharePoint), document management features are critical. However, you will see some minor annoyances, like check-in/check-out functionality becoming available only if certain workflow settings are enabled.

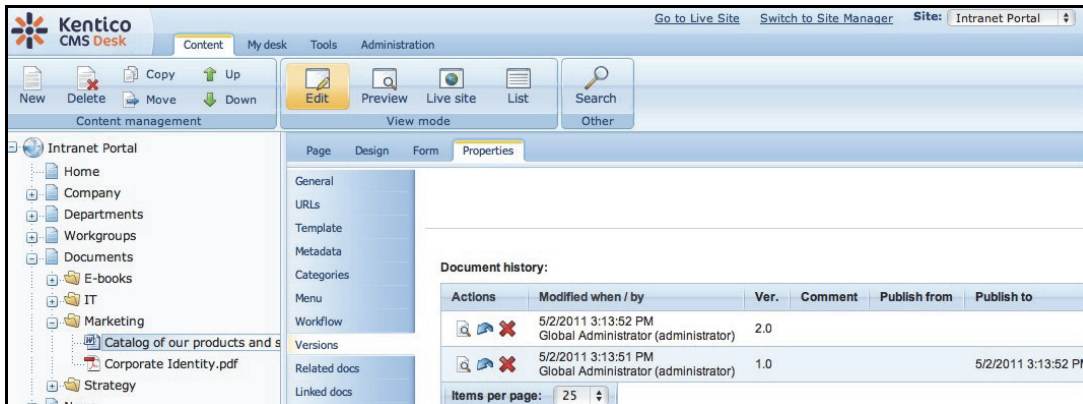


Figure 382. A view into document history and document management in Kentico.

Closely coupled with document management is Kentico’s intranet solution, which provides a default set of simple intranet services. You get the same underlying web content management capabilities, along with workgroups, social interaction, project management, and document management. This is a good starting point for building out custom intranet solutions, as most likely you will need to go outside the starter package that Kentico ships. On the plus side, the Intranet Starter Package already includes a huge variety of pre-built Web Parts, dashboard widgets, and modules (e.g., blog, forum, and calendar). The Kentico CMS SharePoint Connector could also be useful here.

Content Lifecycle

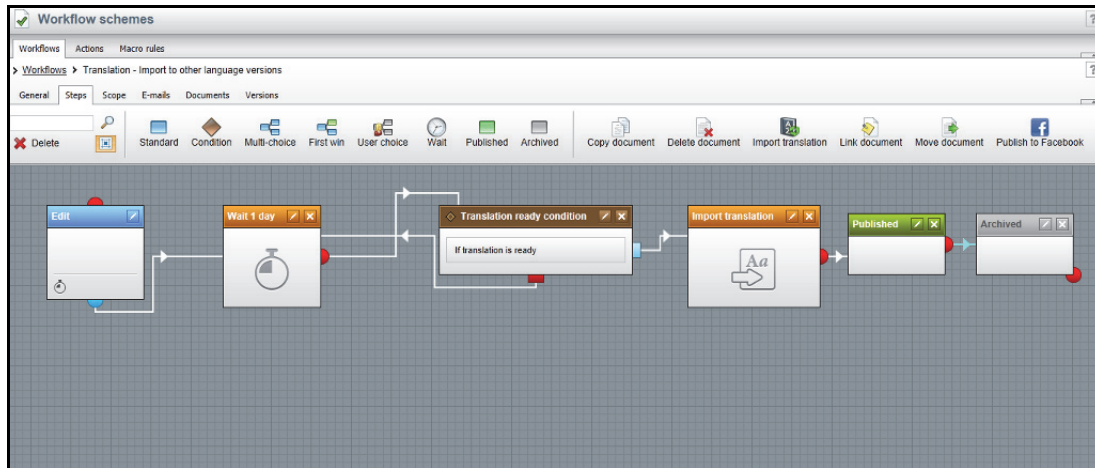


Figure 383. Advanced workflows in Kentico CMS require a great deal of custom code underneath to operate.

The basic workflow in Kentico is a simple sequence of steps (each followed by Approve or Reject). For anything other than the standard workflow (e.g., a simple “Edit,” “Published,” or “Archived”), you will need to use Kentico’s “Advanced Workflow” capability, which provides a visual designer and a number of canned conditions (“steps”) that can be hooked together to create conditional and branched workflows. In order to make the best use of this, you will need to become familiar with the code (“macros”) that supports each of these steps.

This means that for many organizations, in practice, workflow is as basic as it is in DotNetNuke or WordPress; unless you have some great ASP.NET smarts in your enterprise, you should probably forget about complex localization or parallel approval workflows in Kentico CMS. The main reason you may still want to turn on the default workflow is that without it, the system doesn’t have version control, and doesn’t support check-in/check-out locking of content items.

Globalization features are sparse. You can set the audiences available for a specific site, which Kentico calls “cultures” (such as “English - Canada” and “French - Canada”). This means that visitors can select flags on the website and content will be in their language, tailored to their culture. However, Kentico has chosen an odd compromise here; the cultures aren’t exact mirrors of each other (it’s very possible to have pages available in just one culture, and they won’t show in the site navigation of other cultures). However, you’re still expected to create multiple versions of the same document for each culture by creating one, then switching cultures, and creating the other; there are no separate site trees.

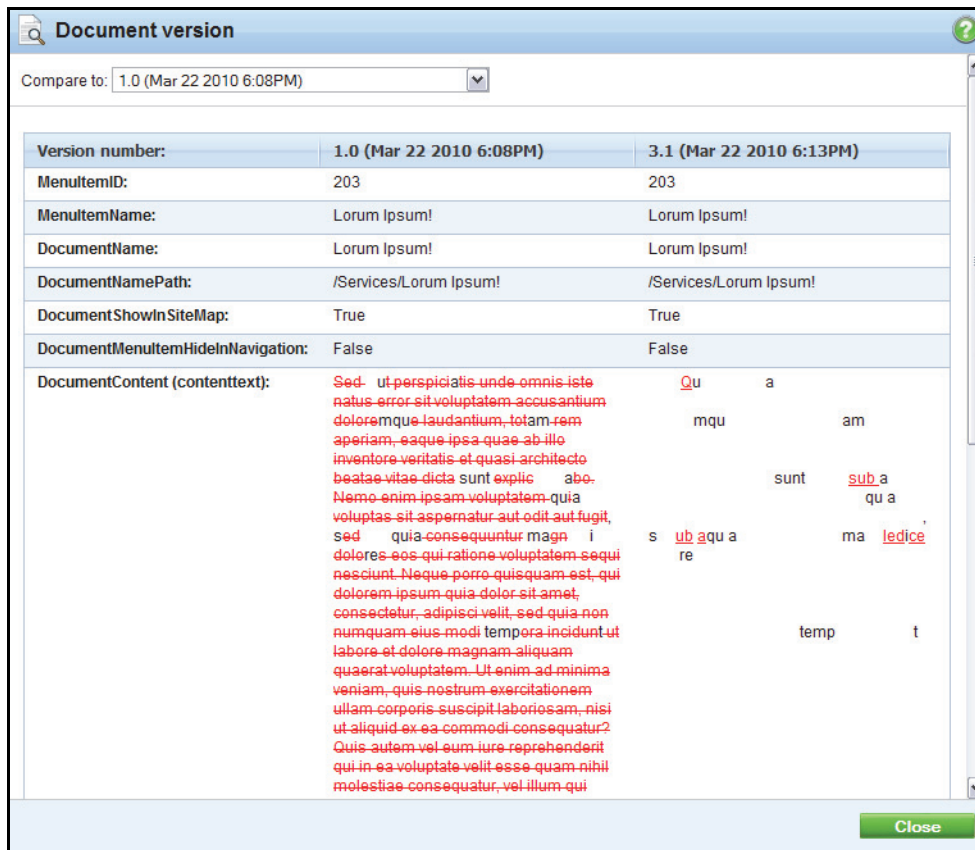


Figure 384. Kentico CMS stores versions if workflow is enabled; the system has a visual compare of versions, but you shouldn't mistake this for a true tracking of changes, as this example demonstrates (between version 1.0 and 3.1, a text by Cicero was replaced by one line from Ovid).

This means that Kentico enables the flexibility of having different sites for each culture (some 150+ are available out of the box), and there is a “fallback” mechanism in the form of the “Combine with Default Language” setting. Of course by using this, you then lose the ability to have different sites, and non-existing translations will default to one language. In practice these supplied cultures are simply containers for language short-codes, which can then be used in instances such as translation workflows or segmentation in the e-marketing tools. For the visitor, this will look like an odd and random mix of languages. This in-between approach may suffice for brochure websites in several languages, but it's too limited for anything more complex than that.

Experience

Publishing

Kentico doesn't seem to have paid particular attention to accessibility of the produced website. This means that it has some of the issues that plague .NET sites in general. Although there's a relatively clean separation between content, controls, and design (applied through stylesheets), there is also an abundant use of JavaScript in the controls. The company notes that accessibility will be determined by your own front-end HTML, and that it offers alternatives to most of the JavaScript Web Parts. If accessibility is a concern for you, test carefully.

Kentico CMS produces friendly URLs, based on the page hierarchy and page names, such as: <http://localhost/KenticoCMS/Services/Lorum-Ipsum!.aspx>

Note that the CMS doesn't sanitize the page names entered by contributors (it will allow any character, which may generate complex URLs that don't fare too well in search engine optimization).

Digital Marketing

Via its optional "Enterprise Marketing Solution" (EMS) Kentico now offers a large number of additional functionalities targeted directly at the digital marketer; from email campaign management, to A/B and Multivariate testing. In general, these functionalities are "wide and shallow," as they are comprehensive in range but basic in form.

Let's review the wide range first. You'll find some contact management capabilities, which can allow population from Salesforce via a basic Force.com integration script. Note that these "contacts" are distinct from "users" that might participate in social activity (e.g., blogs or wikis) elsewhere. Actions attributed to these contacts can then be batched back up to Salesforce — at least in principle, which means that some level of "round-trip" integration with that CRM might be possible.

Experience	
Publishing	
Standards Adherence	
Multichannel	
Mobile	
Digital Marketing	
Site & Campaign Analytics	
Testing & Optimization	
Segmentation & Personalization	
Social Media Integration	
Promotional Campaigns	
Community & UCG	
Workplace	
Collaboration & Networking	
Dashboard	
Ancillary	
Site Search	
Online Forms	
Module Ecosystem	

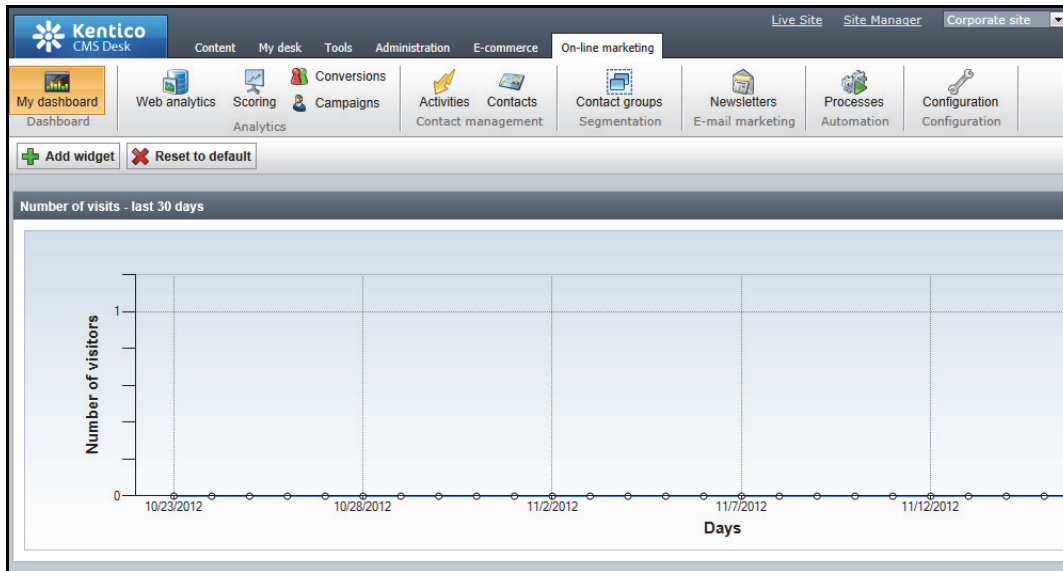


Figure 385. Kentico EMS offers the promise of a great deal of useful functionality.

The A/B and multivariate testing performs reasonably well. Although somewhat illogically placed in the navigation system, the supporting documentation helps create simple tests for both variants of the tests. Again though, you cannot simply connect the tests to segmentations in your “contacts,” which means that testing specific variants on segments smaller than language “cultures” is not possible.

Overall, where the lack of depth within EMS shows is in the lack of integration across each of the features. Each feels distinctly modular and works in isolation from the others; the Email Marketing subscriber lists for example are separate from the aforementioned “contacts” and “users.” Thus, when you create separate silos of people, they may have a significant overlap. Synchronizing these modules together will require considerable organization and effort to be made externally to EMS.

The summary of EMS as it stands in Version 7.0 is that whilst it provides the piece parts of what a contemporary e-marketer might want, wrapped up in a single interface (and license fee) that hides some of the complexity underneath, its collection of disparate modules have little or no knowledge of the other’s existence. To deploy it in its entirety right now would likely require patience and considerable bravery.

Within pages, you can add “widgets” to pages, which are configurable gadgets a CMS user can place in designated areas on the page. These are quite similar to the “Web Parts” used in the page designer described earlier (in fact, technically, “widgets” are special “Web Parts”). However, they can be configured by the end user (the site visitor), who can move them around and configure them (think of iGoogle and its gadgets).

The vendor calls this “simple personalization,” and visitors can use it to build their own homepage. The functionality is quite limited — as is the scope of those who would need this. For many users, designing a page using both Web Parts and widgets is probably too complicated to understand — no matter how easily the interface displays the option. Other than these “widgets,” a do-it-yourself method of personalization by visitors, Kentico has no real personalization features.

To be clear, Kentico CMS comes with a plethora of other potentially useful modules out of the box, including blogs, wikis, a store, an event planner, forms processing, polls, and forums. Here the product has more functionality integrated than most of its competitors, and it's worth noting that with most other systems, each of these modules could be a costly extra. Just understand that Kentico's modules are available in various combinations in the higher licensing options only; the \$1,999 basic license has disabled most of these modules.

Nevertheless, much as with EMS, these modules are of variable capability. For instance, the "e-commerce solution" is very basic. However, the Communities module has niceties such as a "bad word" filter and IP-banning — features that most other systems with social features lack. If you require any of these modules specifically, be sure to test them thoroughly before committing.

If you look into exploring the mobile delivery side of Kentico CMS, prepare yourself to tinker with settings, widget actions, device redirection, and other Web Parts' properties, and other modifications that are mostly expected to be handled more gracefully nowadays.

Ancillary Services

Kentico offers Lucene.NET integration for site search, but this doesn't support all modules (unlike the Java-based Lucene Solr used by many vendors), and it doesn't support faceted navigation of search results).

Vendor Intangibles

Kentico says it has grown organically, without outside investment, and has been profitable ever since its start in 2004. As always, such claims are hard to verify with a privately held company, but Kentico is certainly showing successful growth, both in its number of employees (from four six years ago, to 94 now), and its number of live websites.

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

In an effort to nurture its otherwise rather unloved community, Kentico kicked off various activities to support partners and developers, including a Developer Certification Program, partner certification and regional partner events, conferences in the UK, and the worldwide "Kentico Connection" conferences in Prague and Denver. Since these are all relatively recent developments, you'll need to investigate the quality of your implementation partners carefully before signing any contracts.

The vendor has a large number of partners worldwide (though notably absent in most of South America and Asia). Most of these, however, are relatively small shops. The good news is that it should be relatively easy to find a partner near you; the bad news is that it'll be hard to judge whether they actually have the capacity to be of any use. The main criticisms of Kentico support we have heard are that they have smallish, overstretched integration partners.

As far as its US expansion goes, Kentico is trying. With only eight people so far on the North American grounds, things are not moving as quickly as you as a customer may like to see them move. The opening of a support office in Seattle was significant in helping bridge the gap with otherwise mostly Central European time-based customer support.

Kentico has a good reputation for the quality of its support. The company also states that it will fix bugs within 7 days, along with the tradition of planting a tree for each bug found. For developers, there's a DevNet website, with documentation, tutorials, and how-to videos available online. If there's one caveat here, it's that the level of commitment shown by employees may be hard to sustain, considering Kentico's considerable growth in popularity. Documentation is not always updated in sync with the latest release — especially for new modules and features.

Something to consider is Kentico's overall corporate culture. Some prospects have complained of heavy-handed sales and marketing tactics — the kind you generally see at higher tiers in the WCXM marketplace, but not so much at this level. The company also has a reputation for being a bit prickly in the face of customer complaints. If you like the technology, that might not matter; if you're struggling, it might matter a lot.

The release of the e-marketing-focused EMS version of the product coincided with a simplification of the headline pricing model and there are now only three versions of the product:

- A base level product that excludes all of the advanced options (but retains some e-commerce capabilities) from \$1,999
- An “Ultimate” edition containing all but the e-marketing tools from \$4,499
- EMS itself from \$14,999 (or from \$999 per month via SaaS)

Keep in mind however that these prices are “per domain” and “per server.” Even sub-domains will need to be licensed separately and costs can escalate rapidly — especially if you create “highly available” sites with active/active failover or larger farms.

Kentico CMS has maintained a steady release schedule, with a new major release coming out on average every year and a half. Not surprisingly for a Microsoft-centric system, some of these releases have coincided with major Microsoft innovations.

Conclusion

Kentico CMS comes with a multitude of modules, including most of the social, community, collaboration, and now e-marketing tools you could think of having. However, it lacks the more complex enterprise plumbing (in its very basic repository services), and has only basic categorization and metadata management. This is completely different from Drupal's taxonomy-driven, placeless content publication. However, where Drupal has thousands of modules (most of which are untested), Kentico has all of the important ones — as part of its core system. There may be an overlap in the community and social scenarios that both systems address, but their architecture is entirely different.

Kentico CMS offers a great deal of capability out of the box and although much of that is shallow compared to any enterprise-level alternative, for many smaller organizations, this will provide more than enough depth in which to swim. In some areas however — particularly those central to the value in EMS — the lack of maturity and modular isolation of each part of the solution might hamper even the most nascent of e-marketers.

To a developer, the most comparable systems in this research would probably be Ektron's CMS, EPiServer CMS, DotNetNuke and Telerik Sitefinity. However, Kentico comes in at a lower price than EPiServer for community purposes (keeping in mind the optional modules that EPi requires for that), and it has a cleaner architecture and lower price than Ektron. If you're in the

UK and value accessibility, perhaps you should line up GOSS for a comparison (especially in government environments). Kentico CMS isn't as sophisticated as a Sitecore or SDL for example, but depending on your scenarios, it might make up for that in its implementation ease. Likewise, Kentico might not be as .NET developer-oriented as the lower-priced Telerik Sitefinity, but Kentico the platform and company seem a bit more stable.

Consider Kentico CMS for less complex scenarios, which still require the breadth of features the system offers. However, in its rush to provide all of the functionality any customer could ever want — and all the different ways to implement it — this system risks trying to be everything to everyone, at the same time. This seems to get increasingly worse with each new major release.

The most important thing to keep in mind is that you shouldn't go both left and right in the same implementation, even if the software doesn't stop you from trying to do so. If that fits the bill, your Czech is in the mail.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

MODX: Revolution

modx.com

Vendor at a Glance

Specsheet	MODX: Revolution 2.2.2 Summary
Geography	Global, with emphasis on North America
What's New	<ul style="list-style-type: none"> • "Media Sources" integrates with cloud storage service
Strengths	<ul style="list-style-type: none"> • Flexible, developer-friendly platform is more modern than Joomla! • Employs established PHP toolsets, such as xPDO for object-relational mapping and Smarty templating • Small design agencies find the learning curve is easier than Drupal's • Page-oriented approach works well for sites with simpler content models and limited content reuse • Increasing availability of commercial support for integrators and end users
Weaknesses	<ul style="list-style-type: none"> • Localization services rely heavily on community contributions, resulting in inconsistent quality and limited coverage • Weak directory integration could cause problems for larger implementations • Still a development platform most suited for customers with strong internal IT depts. or relationships with experienced system integrators • No real multisite management services or content inheritance model • Rudimentary workflow capabilities and an absence of versioning services may limit uptake in larger enterprises • User interface is designed for power users and its lack of in-context editing could intimidate casual contributors • Dearth of experienced integrators may reduce your implementation support choices • Community's efforts are divided between two overlapping but incompatible products
Potential Fit	Informational, Microsites
Unlikely Fit	Any Complex or Digital Workplace Scenario
Compare To	WordPress, Plone, Joomla!, Drupal
Operating Systems	Linux x86 / x86-64 Bit, Windows Server, OS X
Repository	MySQL, Microsoft SQL Server
Client	Browser
App Platform	PHP/Apache
Licensing	GNU GPL Open Source License
Ownership	Privately held

Summary

Headquartered in Dallas, Texas, MODX the company maintains two open source WCXM projects; a legacy product “Evolution,” and its recent successor, “Revolution.” “Evolution” — with the last release in 2011 — has gone into maintenance mode, where only critical fixes and a small number of enhancements have been made to maintain the platform. “Revolution” is a ground-up rewrite of the original product, and it reached a stable release in 2010. Now it is the primary focus of MODX and its small army of community developers.

As with all open source projects, the community and ecosystem speak volumes about the project’s overall health. While MODX is no Drupal, there’s substantial activity in the MODX movement, with the number of registered users spiking to tens of thousands.

Although the platform espouses ease of producing new sites, using MODX does presume a fairly high level of competency in HTML, CSS, and JavaScript — even when working with basic elements of the system.

Scenario Fits	
Simpler Site	
Informational	<input checked="" type="checkbox"/>
Microsites & Landing	<input checked="" type="checkbox"/>
Mid-Range	
Basic Digital Marketing	<input checked="" type="checkbox"/>
Mobile Site	<input checked="" type="checkbox"/>
Community Oriented	<input type="checkbox"/>
Complex Site	
Advanced Marketing Portal	<input type="checkbox"/>
Global Enterprise	<input type="checkbox"/>
Multichannel Publishing	<input type="checkbox"/>
Ultra-Large Single	<input type="checkbox"/>
Digital Workplace	
SMB/Departmental	<input type="checkbox"/>
Global/Enterprise	<input type="checkbox"/>

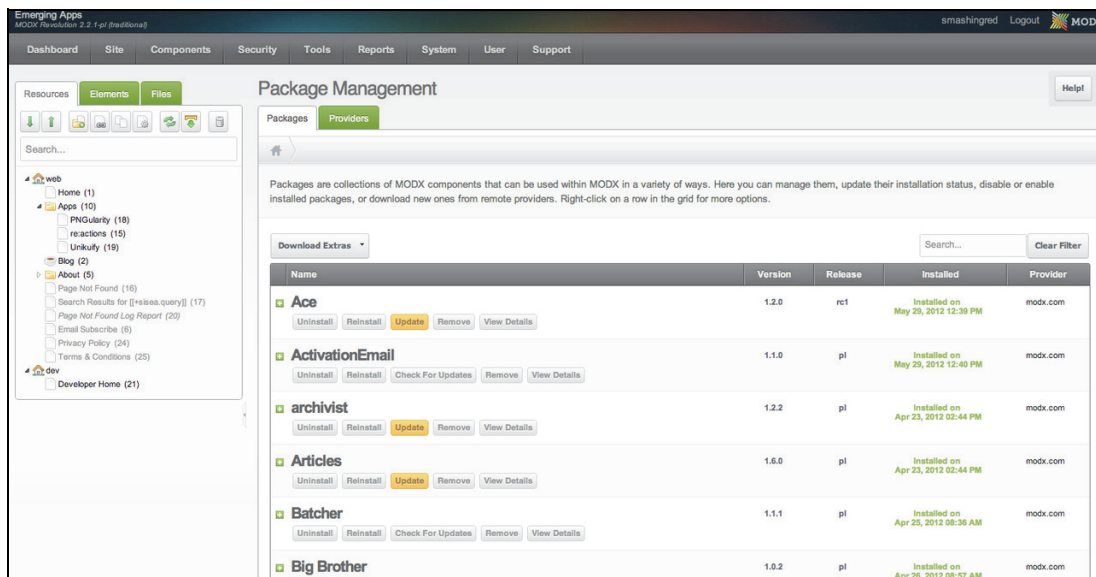


Figure 386. Managing add-ons via the MODX UI.

Note that since MODX has not yet evolved into comparatively as mature an offering as Drupal, you will become dependent on the smallish commercial sponsor (MODX the company), or a distributed community of comparatively small consultancies and indie contractors. MODX’s long-time ambition to launch a more sophisticated, three-tier partner program remains in its infancy.

Without much of a reputation yet, MODX aspires to compete against Joomla!, WordPress and Drupal, facing tough competition and much more established implementations and code bases; it takes a certain kind of customer to find jewels in MODX. Most likely, this customer would be a small web shop, interactive agency, or non-profit organization with decent in-house developers, a penchant for customization, and the savvy to recognize a codebase more modern than Joomla!'s, but lacking the technical chops to wrestle with Drupal.

The company says its older Evolution offering remains apt for smaller sites. We see it as a dying platform, albeit one with a sizable legacy install base. Going forward, you'll want to focus on Revolution.

Introduction

MODX was founded in 2004 by Ryan Thrash with the idea to create a light framework for managing web content, XHTML, CSS, and SEO. The initial code base was written by Raymond Irving as a fork of the Etomite CMS open source project. The forked version was eventually established as the standalone product, Evolution in 2005. In 2010, a second project — essentially a full rewrite of the original — was released under the name Revolution.

While both projects share a common ancestry in PHP, Revolution's development continues at a steady pace (with minor releases on a monthly basis, point releases every 12 months), Evolution's release schedule has naturally slowed to a minor release every six months. MODX targets Evolution primarily for single-domain sites of less than 5,000 documents, while Revolution's greater range of repository support, more flexible caching, and general scalability ostensibly makes it the choice for any multisite or high-traffic/large-content volume scenario.

Technology

Architecture

Evolution's architecture sticks entirely to the LAMP stack. Revolution extends this further by offering support for Microsoft's Server/Database technologies (in its 2008 form) and although this offers the possibility of choice, there is little to suggest that these additional platforms are either widely used or supported by their partner channels. MODX can also run on WAMP, or MAMP, appealing to a variety of organizations using different technology stacks.

Architecturally, Revolution is a dynamic, database-driven “frying” system, where pages are assembled on request from a database (via a Model View Controller — MVC — model) with partial-page caching available to determine which areas within pages are delivered on demand and which are delivered from a front-end cache. As a result, seriously consider the scalability in high-demand scenarios, where a great deal of time is likely required to plan clustering and caching configurations (see Figure 387).

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

Technical Administration & Security

Revolution consists of two primary elements: a content management back-end, and a web application framework for site delivery that sits on top of it. Both of these elements are fairly customizable; the former has enough out-of-the-box support for very basic web publishing, while the latter offers a multiplicity of options for a skilled front-end developer to display this content.

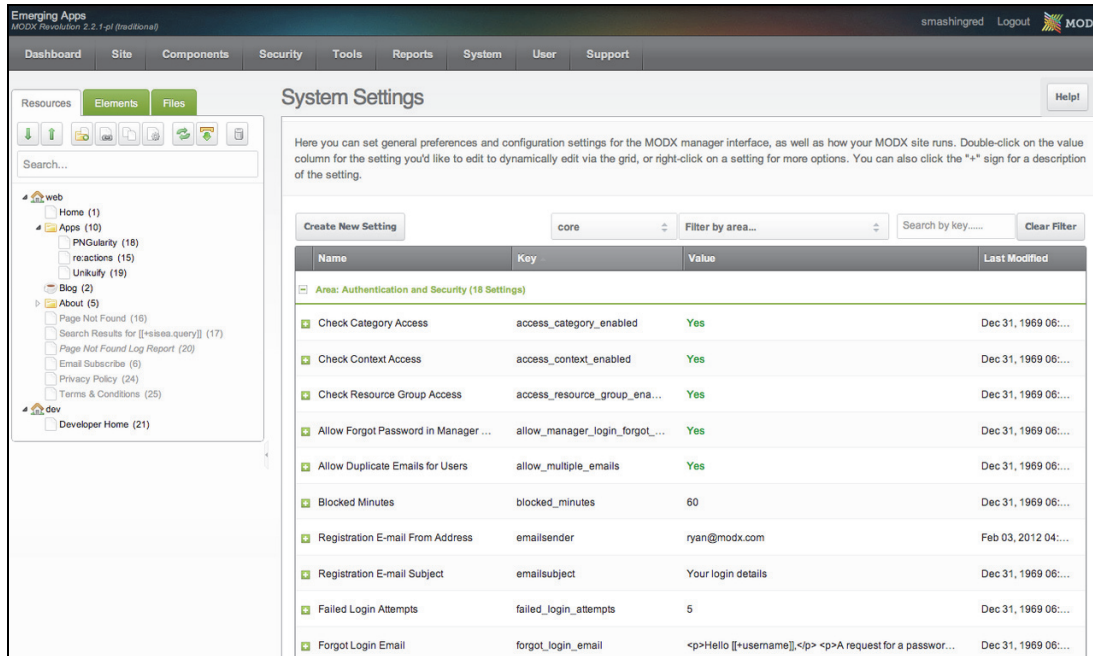


Figure 387. A view into the configuration and system settings UI in MODX Revolution.

Revolution's content management interfaces can be extended if you have strong PHP resources. While there is community support and a wide range of pre-written community extensions that cover many of the additional functionalities, installing, testing, and management still remains within the realm of skilled PHP resources.

MODX security is based on access control lists. Rights and permissions management can become very hairy very quickly due to a very complex design of the security model, creating a task for developers yet again. You will be starting from scratch, so you'll want to work out a sensible structure and policy before implementation.

Note that LDAP and AD integration is available only via an add-on module, and works for authentication only, with no ability to pull permissions/authorizations.

Development

MODX Revolution was built on an object-oriented core. Revolution's object-relational mapping layer takes its roots from a different open source project: the OpenExpedio Project (xPDO). xPDO is one of the better fitting database options for PHP-based content management systems given that its ORB (object-relational bridge) library is compatible with PHP 4 and 5+.

Developing additional elements for the UI should be more straightforward in Revolution, as it utilizes ExtJS framework, although this has come at the apparent cost of load times. Test carefully.

The foundational “building block” in MODX Revolution is a “resource” – a URI, or anything that can be accessed via a URL. A resource can be a static file on a file system, a DB item, or data accessed via Web Services.

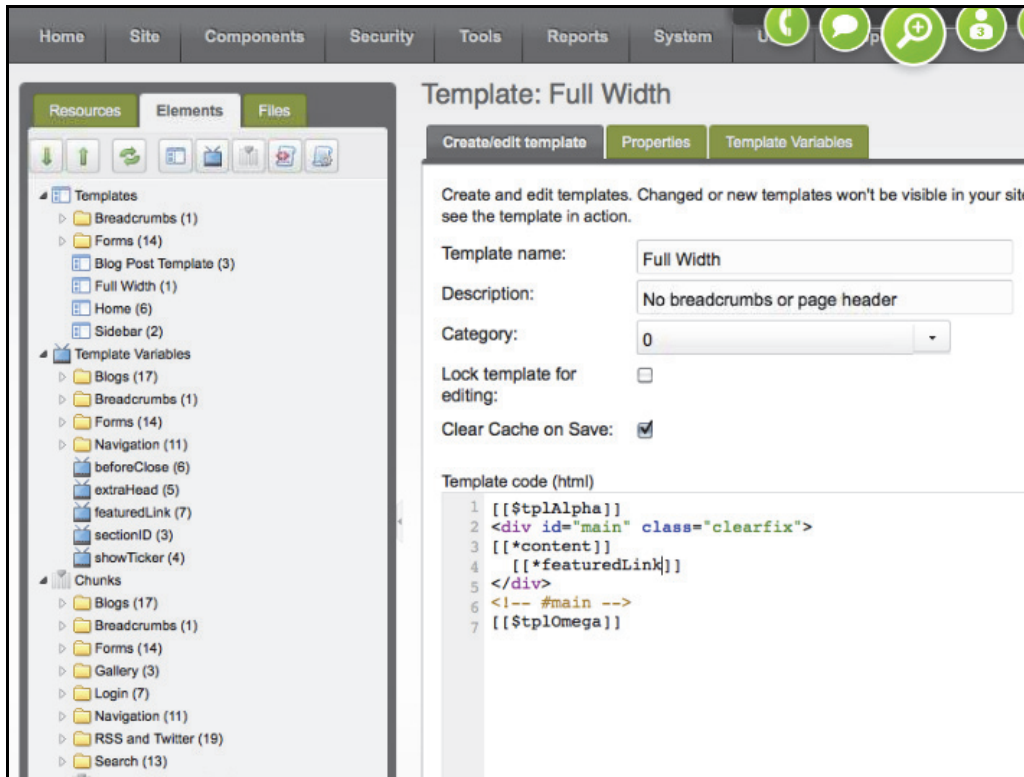


Figure 389. Setting up a new Template Variable in Revolution is no trivial task for those unfamiliar with input types, option values, and output types.

MODX identifies four primary types of resources:

1. Document resource – The element you use on web pages
2. Static resource – Content from the file system
3. Web links – A pointer to other pages in MODX, or an external URL with a 301 redirect
4. Symlink – A link with a 302 redirect to avoid changing the URL

Dynamic linking is a capability present in many WCXM platforms for a number of years now. In MODX, the “Janitor” tool didn’t appear until very recently to provide broken link checking via a MODX native tag syntax. However, MODX claims there’s no need for link checking in its system, because MODX automatically handles link management by generating link tags from cache at runtime. Investigate this area closely if link management is important to you and you want to exercise more control over this area of content management.

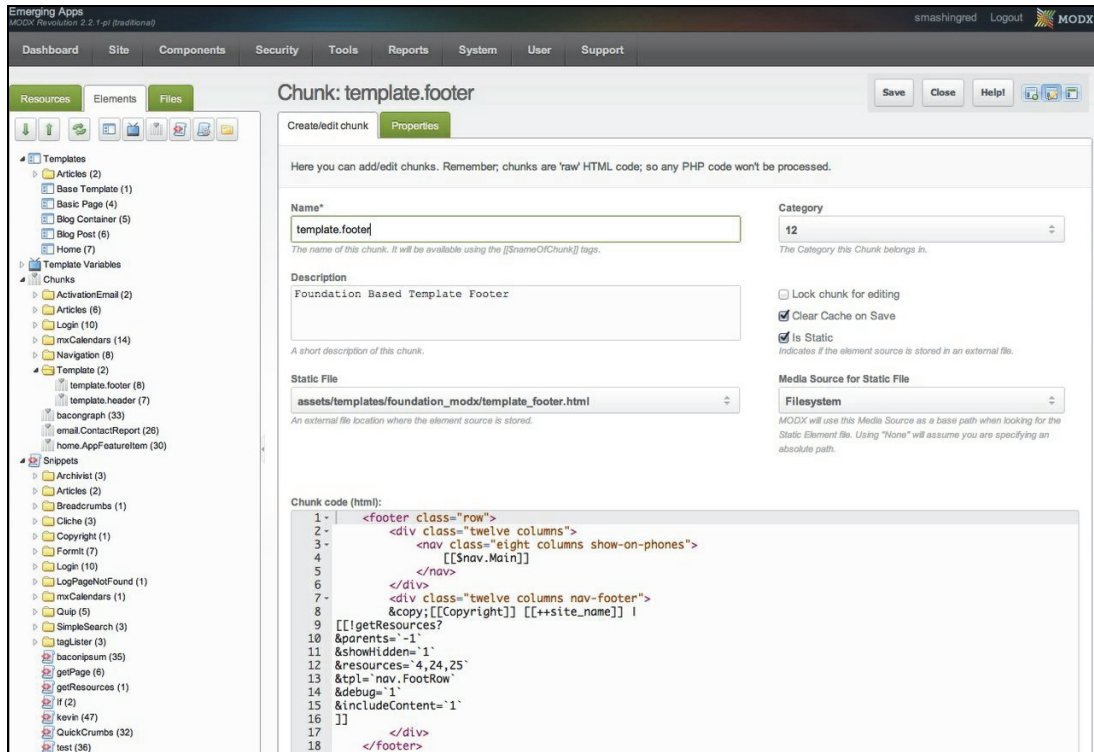


Figure 390. Chunk creation and management in Revolution.

Because MODX is developer-centric, there are a few other terminology items with which users should become familiar. These are elements — `[[$chunks]]`, `[[*tvs]]` and `[[snippets]]`. The limitation of chunks (and thus the need for snippets) is for HTML reuse only. Chunks themselves can contain nested snippets, template variables, or other MODX tags that can run PHP.

Snippets are used to manage dynamic content via raw PHP and/or JavaScript code to render content on the page at request time. Snippets are about the only place in the system where you can separate business logic from layout and presentation. The Wayfinder snippet, for example, manages navigational elements and menus. One word of caution about snippets: there's no native code versioning in MODX Revolution. Therefore, in order to write a new snippet off the existing, live snippet, developers would need to create a duplicate of the older version first (or utilize an external source control mechanism, such as SVN or Git).

A MODX template contains the editable Content Area identified with the syntax `[[*content]]`, and it contains MODX tags that insert snippets. The template also contains Template Variables, which content editors use to insert content or multimedia from outside of the main Content Area.

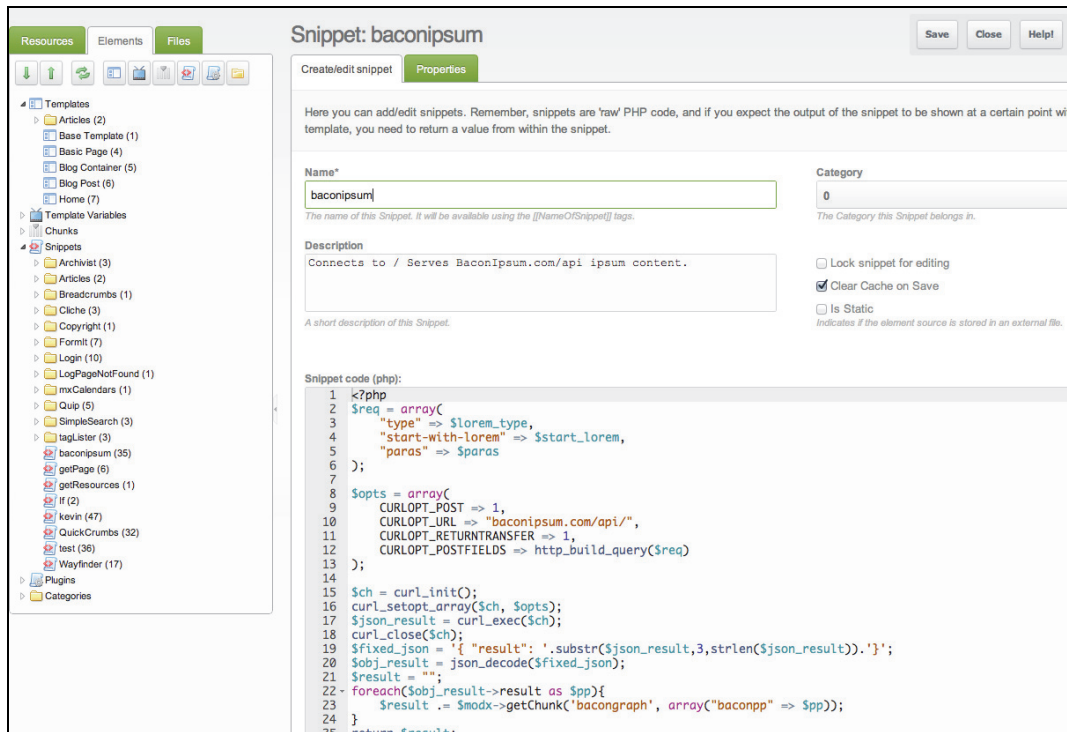


Figure 392. Managing snippets in Revolution CMS.

MODX Tags are used to insert more variable content into pages. Behind the scenes, tags are replaced with the content they output. So, if an editor wants to output a link to another document resource on the MODX site, he/she would use `[[~linkID]]`. The `[[Schunk]]` tag would insert an HTML chunk of code into a page. Developers need to apply MODX tags in order to add content and functionalities to web pages. Since there is no clear separation between editorial and technical tasks, MODX allows tags to be inserted in both content and templates.

Adapting content types themselves requires working directly with the database in most cases. Additionally, while templates (and their associated sub-elements) can be reused, there is no initial way to structure content types for reuse. However, “Template Variables” can be used to add custom fields into content types without adjusting their basic schema, thereby allowing reuse of sub-schemas across different templates.

Despite all of this componentization in terminology, MODX is essentially a page-centric web CMS, where pages get assembled from multiple elements. This limits opportunities to reuse content, given that pages are large chunks of content that are typically harder to reuse than smaller, componentized elements of content, design, and code.

Having struggled with content reuse, community members began to develop their own snippets. As with Drupal, MODX has a variety of add-on modules and extensions that were either built internally by the vendor or submitted by community members through a vetting process. It should be noted, however, that many of these submitted modules are “use at your own risk.”

Offsetting the pains of setting up all these elements and going to a multitude of places to set up various other little things is that it all happens in a single window without the annoying pop-ups that prevail in many other WCM systems. On the downside, this creates a lot of vertical space and scrolling.

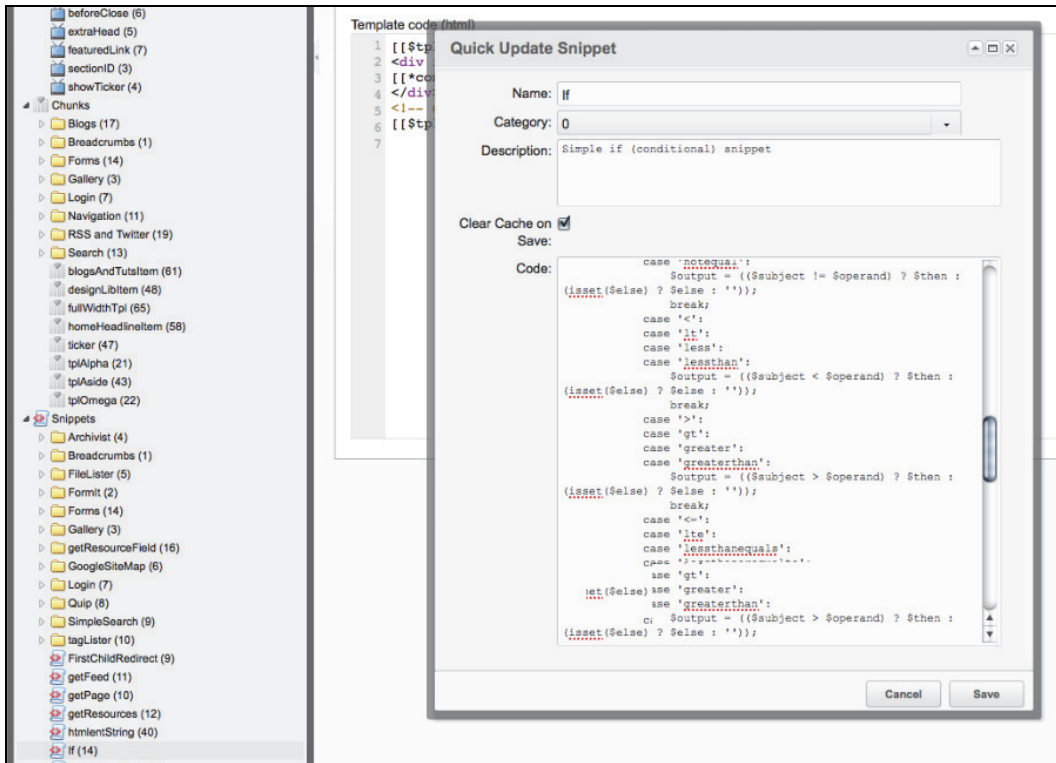


Figure 393. While there aren't too many pop-ups, there are some — such as when you edit raw PHP code snippets.

Templates are written using Smarty, a common PHP development scripting language. Revolution ties templates and CSS together at a code level, which requires clever management of classes to avoid potentially large CSS overheads. CSS management is handled outside the Content Manager via an appropriate design tool.

Performance

Historically, MODX has had scalability issues with its legacy Evolution product, limiting it to sites of around 5,000 assets. Revolution is based upon a new Model View Controller that (in theory) allows you to maintain sites of much larger scale. By default, Revolution has a caching model switched on (which can be further refined by using the “partial page” mechanism), meaning that once the initial cache for each page/site is built, requests for pages are delivered from there, rather than by dynamic generation.

Still, Revolution follows a dynamic publishing model and this generally requires careful management and focus toward anticipated usage — not only visitors, but also contributors. Both classes of access will share resources (back-end and front-end systems are tightly

coupled). Pay attention both to load testing and overall cache performance during evaluation against anticipated user/visitor metrics.

```
01 $config options = array (  
02     /* define the mutability of the master db configured during install */  
03     xPDO::OPT CONN MUTABLE => true,  
04     /* define an array (or JSON string) of read-only slave connections */  
05     xPDO::OPT CONNECTIONS => array(  
06         array(  
07             'dsn' =>  
08             'mysql:host=127.0.0.1:19571;dbname=repo replication;charset=utf8',  
09             'username' => 'rsandbox',  
10             'password' => 'rsandbox',  
11             'options' => array(xPDO::OPT CONN MUTABLE => false),  
12             'driverOptions' => array(),  
13         ),  
14         array(  
15             'dsn' =>  
16             'mysql:host=127.0.0.1:19572;dbname=repo replication;charset=utf8',  
17             'username' => 'rsandbox',  
18             'password' => 'rsandbox',  
19             'options' => array(xPDO::OPT CONN MUTABLE => false),  
20             'driverOptions' => array(),  
21         ),  
22     /* define options for the initial connection to be made, i.e. make a read-only  
23     connection */  
24     xPDO::OPT CONN INIT => array(xPDO::OPT CONN MUTABLE => false)  
25 );
```

Figure 395. MySQL Replication in MODX Revolution 2.2.

Revolution's MVC stores session data at database level, which means that there is a degree of freedom when load-leveling your web servers, since each user does not need to persist on an individual machine for the duration. The latest version of Revolution additionally supports MySQL database replication, which adds further options with database balancing and removes what was previously a bottleneck within the architecture.

Naturally, this means that data repositories not only must be well balanced, they also should be properly hardened against attack. Clever as this delivery tier is, it is not for the technically faint of heart, and setup and ongoing maintenance will require a high level of systems administration skills — beyond what is typically found in many IT organizations.

Little support is provided to support these tasks out of the box: just a simple cache clearance mechanism. It is a skilled development task to configure cache configurations or the model's logic itself; it should be undertaken within an IDE.

Content

Contributor Experience

The UI has been completely overhauled in the Revolution rewrite, but some core elements persist in how the Content Manager flows, albeit with “Elements” and “Resources” nested separately. This could aid user migration from Evolution to Revolution.

MODX’s usability shortcomings for the non-technical crowd stem from its original intent to manage content and presentation using XHTML and CSS — a world unknown to the majority of marketers and content authors. The UI is designed for power users and may need “dumbing down” for casual content contributors.

Contributing Content

Content editors work primarily with resources within Content Areas and with the rich text editor in the MODX Manager, utilizing predefined Template Variables.

Content	
Contributor Experience	
Overall Usability	<input type="checkbox"/>
UI Accessibility	<input type="checkbox"/>
Contributing Content	
Authoring & Transformation	<input type="checkbox"/>
Tagging & Taxonomy	<input type="checkbox"/>
Content Reuse	<input type="checkbox"/>
Media & Document Management	<input type="checkbox"/>
Repository Services	<input type="checkbox"/>
Content Lifecycle	
Workflow	<input type="checkbox"/>
Globalization	<input type="checkbox"/>
Archiving & Compliance	<input type="checkbox"/>

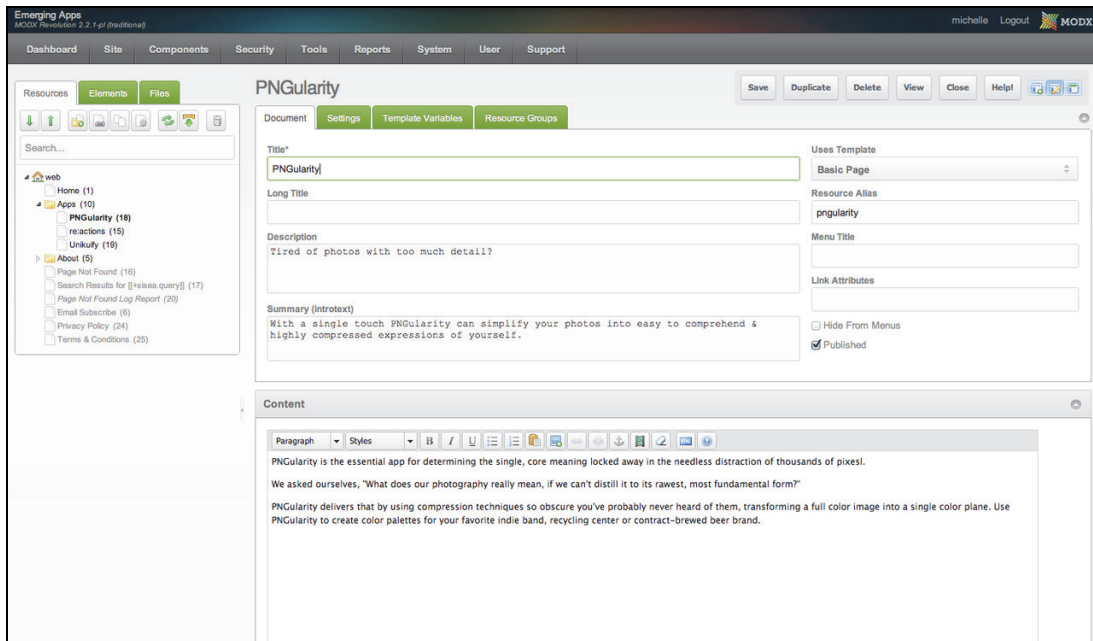


Figure 396. Revolution Content Manager (composite) editing panel.

A MODX template contains the editable Content Area identified with the syntax `[[*content]]`, and it contains MODX tags that insert snippets. The template also contains Template Variables, which content editors use to insert content or multimedia from outside of the main Content Area.

It is not unusual that content editors working in MODX CMS have to learn how to manipulate predefined content or code components, as well as get a handle on inserting chunks into Content Areas or Template Variables' there are all tasks that require varying degrees of technical skills.

Revolution utilizes the common TinyMCE for rich text editing, but you can plug in other RTEs as well. Note that one of the most glaring shortcomings of the system on the editorial side is its lack of in-context editing.

Beyond the simplest of contribution tasks undertaken via TinyMCE, some knowledge of underpinning web technologies becomes increasingly important for users. Like other open source platforms, content contributors face a steep learning curve beyond the basics. You can mitigate some of this by deploying appropriate "add ons," but it is important to consider the training and support that might be required for non-technical users to become proficient with the platform.

New in the latest version is the concept of "Media Sources," which allows external storage locations (e.g., cloud resources like Amazon S3 or Dropbox) to be added to the file tree via a template variable. This adds a large degree of flexibility to asset management, especially if you possess large amounts of existing assets that you wish to reuse in existing repositories. The security implications of these external dependencies need to be carefully considered, as it is likely that access to them will be on a strict inclusion/exclusion basis. While this security of a sort, anyone requiring some asset-level security will need to look beyond this feature.

Document and Media Management

Multimedia content management is very light, with basic tools for very simple placement of pre-formatted images into pages. Revolution CMS does not ship with image tools for cropping or resizing by default, so these tasks must be performed outside the CMS. Extras are available within the community to fill some of these gaps: "php Thumb Of" is a thumbnail generator that seems particularly popular with MODX users. Overall however, there is little to suggest that the platform is suitable for anything more than very basic image work.

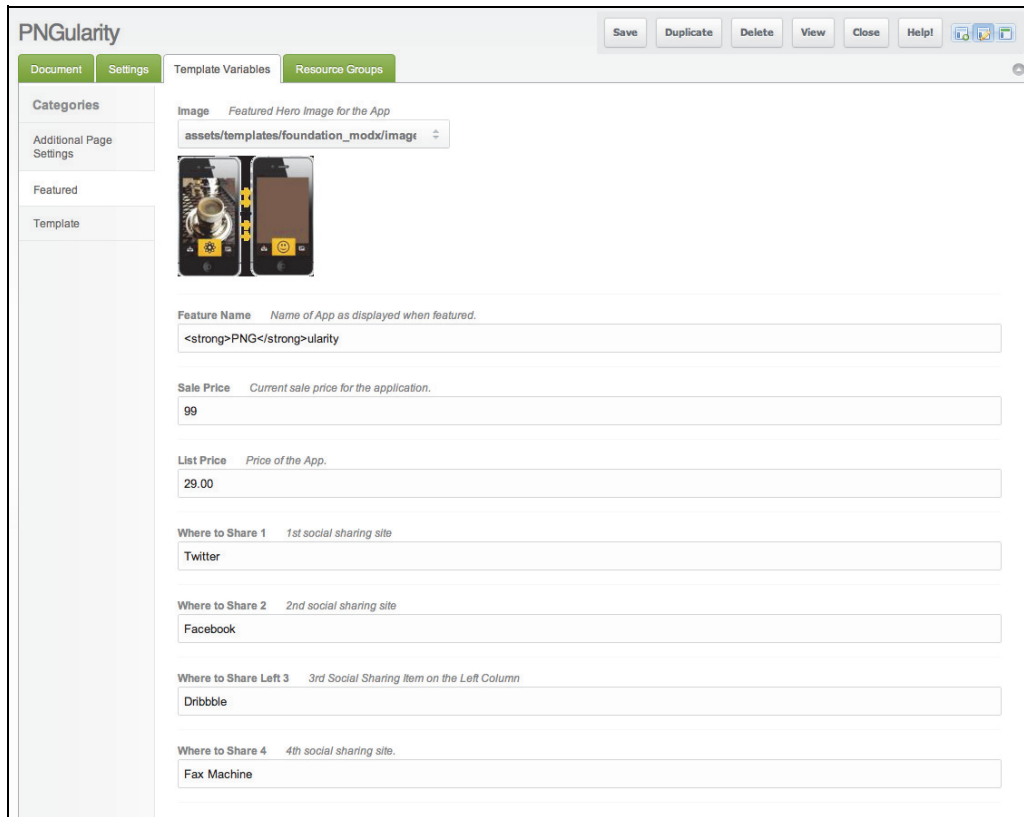


Figure 397. Multimedia content management in MODX's CMS is very light, with basic tools for very simple placement of pre-formatted images into pages.

Video is also left for the add-on community to support, with nothing more than basic embedding and display modules currently offered. If video support beyond simply plugging in YouTube or Vimeo content is central to your requirements, some heavy development work likely will stand between you and a sufficient MODX implementation.

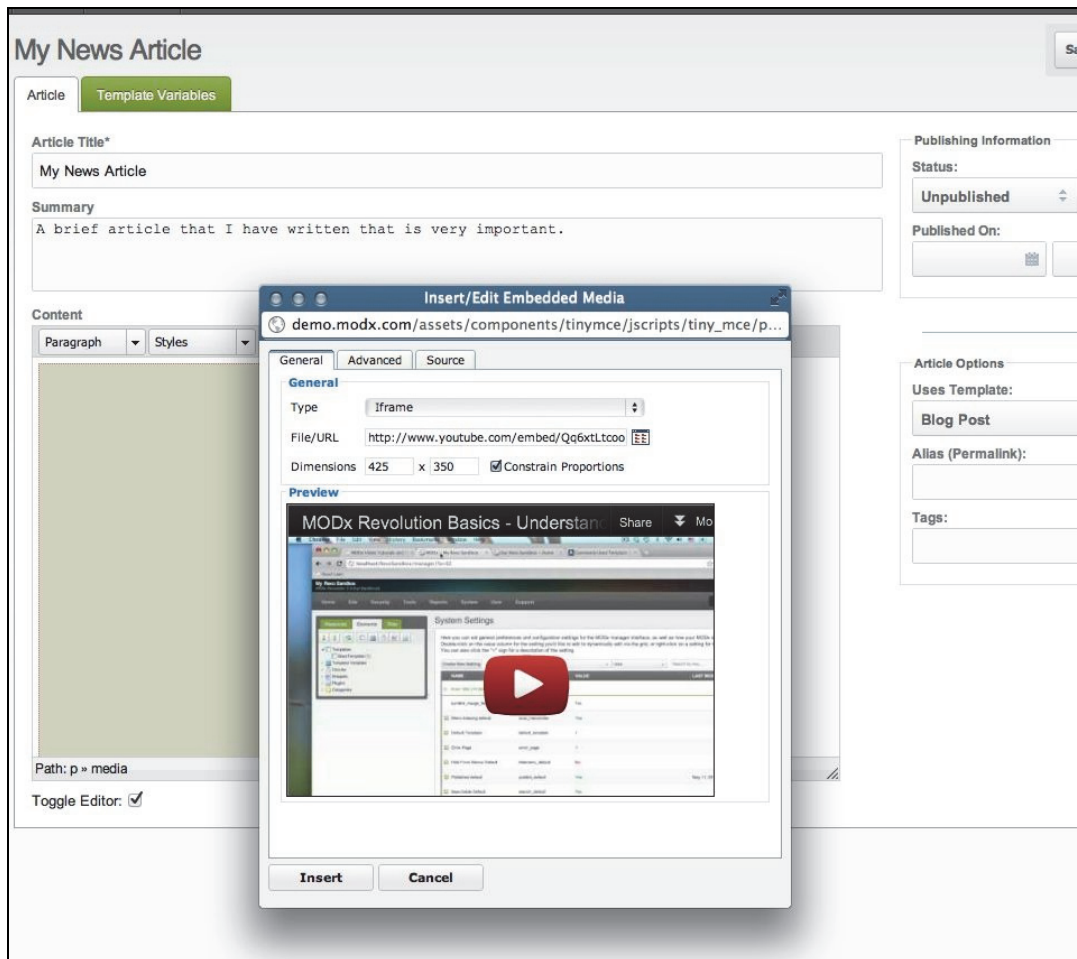


Figure 398. Revolution video management does not extend further than the ability to embed YouTube links into pages.

Content Lifecycle

Revolution offers very little in terms of lifecycle functionality. Versioning and rollback of content items is not present in any form within the product. While some user activity gets recorded within the repository, you need to deploy an add-on utility tool to gain much insight into this information.

Similarly, workflow is basic — similar to WordPress — where content can be drafted and deployed in a basic binary on/off manner. Additional workflow tools can be sourced from the community, but looking at download statistics for such extras, they seem well down the priority list for Revolution implementations, which suggests that it is not being deployed in regulated environments or in environments with large numbers of content editors.

Scheduled deployment / un-deployment is supported within the core of Revolution as part of the default “Resource Settings”

Overall, content governance remains very weak.

Experience

Publishing

MODX has the concept of “Contexts” to support multiple sites — primarily sub-domains — within a single instance of Revolution. This is targeted toward multisite and multilingual scenarios, where configurations and systems resources may be shared across a number of site/language variants based upon an original master version (or localized as required). Sharing content between these variants is possible, but it’s far from a simple exercise. For example, creating a symlink between different contexts requires some application source code hacking. It’s best to keep a courageous PHP developer close at hand if you try to push Revolution in this direction.

Friendly URLs — essential for good SEO practice — can be supported using one of a number of add-ons available within the MODX community. Of course, if you are planning to use Apache as your serving technology, you can fall back on the “mod_rewrite” workaround that you may already know from projects like Joomla!

With respect to mobile, some add-ons provide device detection and re-direction toward mobile-specific templates.

Digital Marketing

MODX Revolution provides little by way of digital marketing, preferring to leave the production and maintenance of those features to the “add-on” community.

Segmentation services are available via add-ons that detect users’ status and deliver different “chunks” of content accordingly. As such, these are really only suited to very explicit scenarios. More advanced implicit modeling — i.e., modeling based on anonymous user click behavior — requires custom work. The community also supplies add-ons that allow for Facebook’s Connect API for social registration and social login.

Google Analytics is another popular area of support within the MODX community. While it is entirely possible to use another provider, there is currently no add-on support beyond Google’s offering. Available add-ons include those designed for automatic deployment of JavaScript tracking code and interface additions where statistics can be viewed directly within the Revolution dashboard.

Experience	
Publishing	
Standards Adherence	
Multichannel	
Mobile	
Digital Marketing	
Site & Campaign Analytics	
Testing & Optimization	
Segmentation & Personalization	
Social Media Integration	
Promotional Campaigns	
Community & UCG	
Workplace	
Collaboration & Networking	
Dashboard	
Ancillary	
Site Search	
Online Forms	
Module Ecosystem	

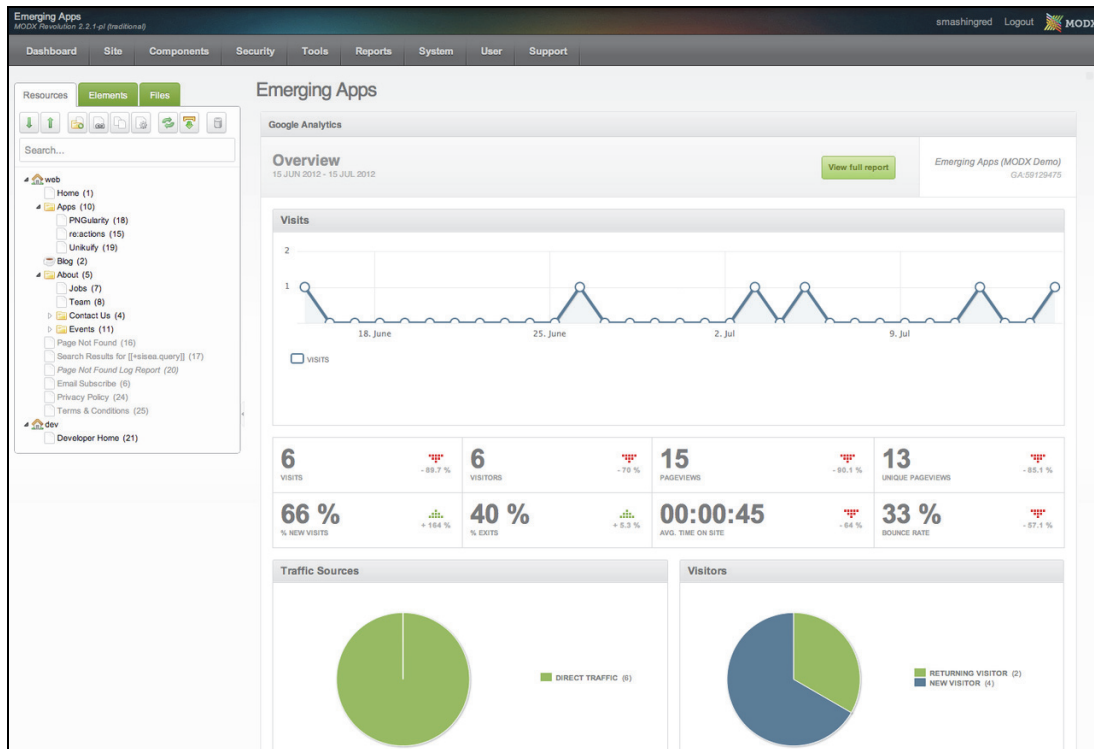


Figure 399. Google Analytics dashboard within the MODX Revolution UI.

Commenting represents a comparatively complete (and therefore popular) offering within the MODX community. The “Quip,” plugin that offers comment moderation, discussion threading, and email notifications has been downloaded over 20,000 times.

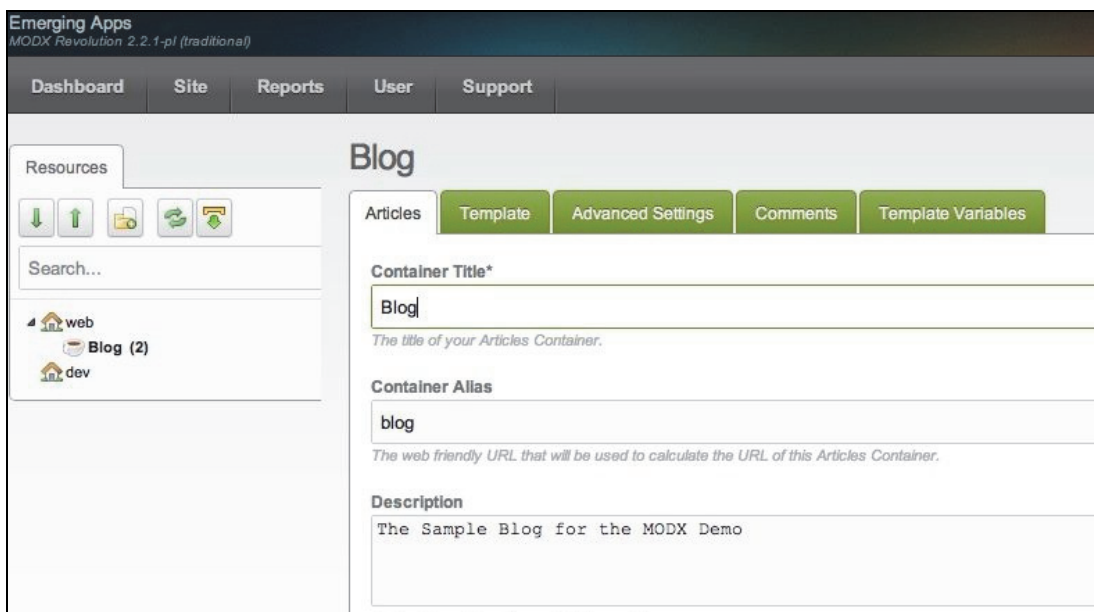


Figure 401. The MODX “Articles” add-on can be used to create blog entries.

New in 2.2.2 of MODX is the “Articles” add-on, which is a blogging capability, very similar in scope to that you would find in more dedicated blogging projects such as WordPress. As such, this also may be suitable for basic news and marketing information, and MODX has import wizards for competitive platforms such as WordPress.

Ancillary Services

Site search is one area where the product has been weak. Going back to the Evolution days, the primary mechanism for adding this functionality has been to add the “Simple Search” module, which provides 2 basic “snippets”: a search form and a search. To extend search capabilities into “faceted search,” you’ll need to befriend various system and configuration hooks, which allow you to add search results via snippets from other sources. Again, keep a developer handy for this.

An additional module can extend “Simple Search” to use Solr as the primary index, which potentially could greatly expand your possibilities. However, this requires a great deal of local Solr smarts (and experience) to implement, putting it beyond all but the most technically proficient organizations.

Project Intangibles

The vast majority of the existing installed base is still on the legacy Evolution product. This is not indicative of any specific issues with Revolution per se, but be aware of the familiarity that much of the existing SI/Integrator/Partner channel has with that product and its suitability for the small/medium projects on which they are focused.

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

MODX is not the first organization to maintain two separate and distinct WCM offerings with considerable potential overlap. The bulk of the install base remains firmly in the legacy Evolution camp. Given the deliberate lack of compatibility between the two projects, moving from Evolution to Revolution falls very much into the area of migration rather than upgrade. Continuing to maintain and support both projects — although Evolution maintenance is now limited to security updates — could stretch MODX, its limited partner base, and community supporters in the medium term.

The Revolution roadmap carries the tag line “All [dates are] dependent on community involvement,” which albeit transparent, might be a bit alarming for those who await the delivery of a critical piece of functionality.

MODX is beginning to offer a number of services directly to customers and integrators in support and custom development.

For SIs, MODX offers a tiered subscription package (“MODX Complete for Agencies”) based around the number of hours required per month, ranging from an initial 8 hours, through to a permitted maximum of 16. This is an interesting offering, given the relative paucity of skilled SIs in many regions of the world, and may encourage more integrators to look into Revolution. Note that MODX also competes with integrators by offering a custom application development service.

For customers, MODX offers a similar “MODX Complete for Enterprise,” with a promised two-hour response within North America and Europe. Costs are also likely to be based on the

number of hours required. Unlike the agency offering, prices are not publicly available. Additionally, for those requiring support without an existing support contract, an “Emergency Support Service” is available and is priced on an hourly basis.

On the whole, MODX's commercial support offering remains very much a second-level (i.e., advanced) offering. The company clearly expects that the bulk of support issues will either be handled by the implementer, or via their busy and well-used forums.

MODX has a fairly active development community, which contributes not only code, but also a number of add-ons that extend the base functionality of the product for front- and back-office requirements. This ability to simply add extra functionality is useful, and many of the most popular and frequently downloaded “add-ons” seem to fill the chasm between Revolution’s “out-of-the-box” product and most commercially licensed competitors.

Conclusion

MODX has made the transition from its origins as a forked version of an existing community project, to producing a more-than-credible open source, second-generation development project. Assess carefully whether the ultimate cost and complexity required to turn Revolution into a fully featured, end-user friendly content management platform is outweighed by the potential flexibility and low-entry cost that it undoubtedly offers. This can be said for many community-oriented products, but the issue is more complicated in this case, because the bare, out-of-the-box capabilities are so limited (e.g., no versioning), but the potential capabilities of the platform are apparently so large.

With Revolution today this puts you in a particular dilemma: you will likely need tech help to get up and running, but MODX’s partner channel is still comparatively weak — in terms of depth and geographical coverage. On the other hand, if your firm favors smaller, more informal development shops or indie contractors, you may find a good partner match.

As with any platform offering a multitude of “extras” and “add-ons” developed by community members, undertake serious diligence to understand and separate the core Revolution product from add-on extensions of varying provenance and support.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

OmniUpdate: OU Campus

omniupdate.com

Vendor at a Glance

Specsheet	OmniUpdate: OU Campus 9.17 Summary
Geography	USA and Canada
What's New	<ul style="list-style-type: none"> • Automatic XML sitemap creation • Accessibility testing module • “Multi-Target Publish” deployment support
Strengths	<ul style="list-style-type: none"> • Available most commonly as SaaS • Focused on higher education, with many references in that sector • Decoupled architecture and page-based orientation will appeal to universities with simple websites • Good accessibility controls and multi-browser preview services • Company puts new features in production on a frequent basis • Well-regarded user group adds good value to the offering, especially for higher-education customers • The UI and product features seem optimized for infrequent, non-technical contributors
Weaknesses	<ul style="list-style-type: none"> • Workflow supports simple approvals only • Minimal support for true multisite management • Dearth of interactive services — favors mostly static publishing • Not well suited for component content reuse • Comparatively poor support for managing images and documents; it's not a good fit for most intranets • May be experiencing growing pains with a comparatively small staff of only around 60 employees • User interface is outdated and is only available in English • On-premise version is essentially an appliance requiring remote administration by OmniUpdate
Potential Fit	Informational Sites, Community Oriented
Unlikely Fit	Digital Workplace, Global Enterprise, Ultra-Large Single Sites
Compare To	CrownPeak, Upland Software, Hannon Hill, Ingeniux, TERMINALFOUR
Operating Systems	Linux, Solaris, Windows for on-premise Enterprise Server
Repository	On-premise bundles Sybase; also supports MySQL & MS SQL Servers
Client	IE, Firefox, Safari, Chrome in Windows, Mac, and Linux
App Platform	Software as a Service; optional Java, on-premise appliance with embedded database and application server
Licensing	Subscription-based, starting at US \$21,000/year
Ownership	Privately held

Summary

OmniUpdate is a Web CMS vendor focused on higher education. Based on the Software-as-a-Service (SaaS) model, OmniUpdate offers a comparatively simple interface for the different types of users in the educational market.

An important point to note is that even though the company provides the software in a SaaS model, the delivery environment always resides with the customer and hence, the complete system is always a combination of two environments: one as a SaaS model (for content production) provided by OmniUpdate, and the second one managed by you, the customer. You can always outsource that to another SaaS vendor, but it’s more streamlined to have one vendor manage all of your needs.

The company (and its customers) claims that OmniUpdate truly understands the vagaries of higher education, and the company boasts that the product supports more than 550 higher-education websites. While the company may understand this niche, clearly part of their success has come from being “omnipresent” at higher education web and computing conferences, as well as the power of word-of-mouth in that sector.

Consider OmniUpdate for simpler scenarios, featuring a decoupled architecture, where SaaS architecture and static or semi-static file publishing meets your requirements.

Scenario Fits	
Simpler Site	
Informational	<input checked="" type="checkbox"/>
Microsites & Landing	<input checked="" type="checkbox"/>
Mid-Range	
Basic Digital Marketing	<input type="checkbox"/>
Mobile Site	<input checked="" type="checkbox"/>
Community Oriented	<input checked="" type="checkbox"/>
Complex Site	
Advanced Marketing Portal	<input type="checkbox"/>
Global Enterprise	<input type="checkbox"/>
Multichannel Publishing	<input type="checkbox"/>
Ultra-Large Single	<input type="checkbox"/>
Digital Workplace	
SMB/Departmental	<input type="checkbox"/>
Global/Enterprise	<input type="checkbox"/>

Introduction

OmniUpdate the company dates all the way back to 1982, making it one of the oldest vendors in this report. Back then, it was called “Mainstay” and focused on publishing Macintosh software.

In 1999, the company changed its name to WebsiteASP, indicative of what is now called SaaS. Finally, in 2006, the company changed it to OmniUpdate, and pinpointed higher education as its prime market. OmniUpdate is based in Camarillo, California — just north of Los Angeles — and has about 60 employees, making it one of the smallest commercial vendors in this report.

OmniUpdate released version v9.17 of OU Campus Web CMS in November 2012. The “dot-seventeen” is revealing; the company frequently releases minor updates, primarily to add new features — at times as many as 50+ new features across a span of several releases. The rapid-release pace has turned some customers off, but others appreciate that OmniUpdate seems intently focused on customers’ feature requests.

Technology

Technical Administration and Security

OU Campus is a J2EE application that runs in a servlet container (such as Tomcat). It also needs a database for storing configuration detail, such as access permissions and toolbar customizations. OU comes bundled with Sybase ASE database, whereas content itself is stored as flat XML files on the OU Campus staging server.

Most customers use the product via SaaS, managing their content on OmniUpdate servers and then deploying it back to their own website environments.

OU Campus is also available as a standalone installation. According to OmniUpdate, about 10 percent of their new customers choose this on-premise version, up from 5 percent a couple of years ago. The installed version is more or less a black box, with an embedded database (Sybase Enterprise) and appserver (Tomcat). OU Campus Web CMS offers a database abstraction layer, which enables you to use other databases like MySQL and Microsoft SQL Server. If you are doing an in-premise installation, you can also use any other Java application server.

For most scenarios, OmniUpdate typically installs and upgrades these on-premise implementations remotely, making this version of the offering a quasi-appliance. You'll need to get comfortable with that, especially since the SaaS-oriented agile development model requires very frequent updates. They also provide you with manuals in case you prefer to do this on your own, but don't consider OU Campus a development platform like some pricier (and more complex) solutions in mid-range and upper-range tiers.

As a decoupled system, OU Campus does not handle content delivery. It publishes static or semi-static files. You can set the system up to move those files via FTP or SFTP to a campus data center. Note that there is no services-oriented access that goes directly to the OU Campus repository; it's completely a push model. In the most recent update, OmniUpdate added what it describes as "Multi-Target Publishing," which is aimed at improving the historically weak content deployment process. Access to each, individual publishing target is controlled by permissions (e.g., it should be possible to restrict users to only publish to a staging site for QA purposes).

Depending on your requirements, you can set up OU Campus to publish to any templating language (e.g., JSP, ASPX and PHP), which enables you to wire in dynamic functionality to the pages. However, OU Campus will only natively virtualize PHP, so for all others you must push them to a staging server via "Multi-Target Publish" to preview them properly, or use some scripting to display dynamic content from the production server directly. That's a hassle that most coupled systems typically avoid. You can also publish to XML, which gives you most flexibility in terms of applying presentation to your content, depending on delivery destination. Fortunately, you can publish to multiple formats (e.g., HTML, XML, and PDF)

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

simultaneously, and use a multi-output preview functionality to see how the content will look after it's published.

OmniUpdate also provides a potential alternative to the decoupled architecture via an add-on module called "Live Delivery Platform," which is installed on the customers' production environment. In principle, this would give you the option to use OU Campus as both a CMS and web platform (similar to the way that Upland Software provides a coupled service). However at this point, the only option "Live Delivery Platform" has materialized in is a server-side, add-on module for the creation and management of forms, surveys, and polls — a far cry from a delivery platform in a true sense.

You can schedule publishing using a wizard. After you've configured scheduled publishing, the wizard asks if you want an email reminder, and it also provides you with the option of scheduling the expiration — quite nice.

While publishing is quite flexible, you should spend a good amount of time testing it. Often in decoupled environments, it can be tricky to ensure that features such as incremental publishing and content expiration on production environment function properly.

You can use Active Directory records or connect via LDAP for authentication to the editorial environment. However if you don't use LDAP, the product has its own online user repository, which can be configured through the browser. OmniUpdate also offers single sign-on through CAS and Shibboleth.

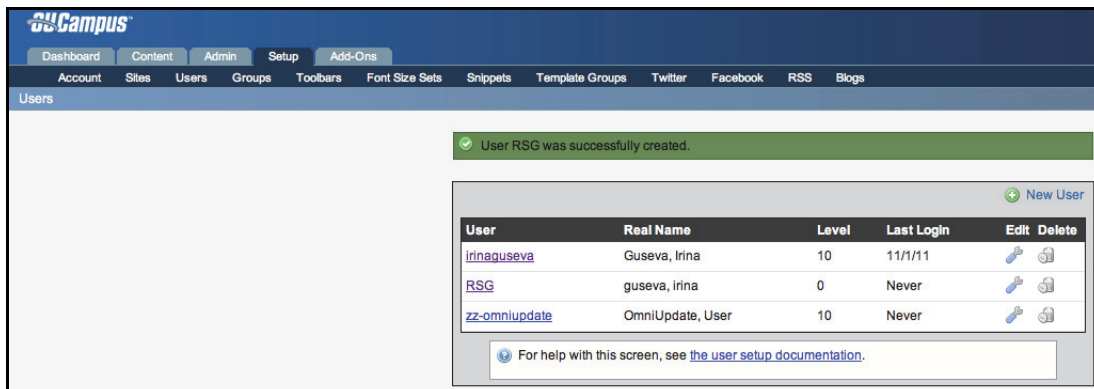


Figure 402. You can manage users, groups, and their access levels using the Users panel in the Setup interface.

Access is controlled via a combination of users, groups and authority levels (Figure 402). Once you create users, you assign them a level that defines what they are able to do within the system. The different levels are Reviewer, Contributor (with four finer-grained levels of access nested within), Editor (with three finer-grained levels of access nested within), Designer, Assistant Admin, and Admin. In addition, there's a separate Approver level for page approval, but it can be overridden on a single-page basis.

The platform includes a set of pre-canned reports. You can also create custom reports using a wizard-like interface, by selecting different query parameters such as categories and data ranges. Unfortunately, you can't click on the query results in all reports; in some cases, you get reports, but no action panel. One available action is to export the report in the .csv format.

Address	Dependency Tag	Directory Variables
/welcome	{{d:6908097}}	section
/snippets	{{d:6908096}}	N/A
/properties	{{d:6908095}}	N/A
/press_releases/images	{{d:6918003}}	N/A
/press_releases	{{d:6908094}}	N/A
/onepage	{{d:6908093}}	section
/OMNI-RESOURCES/dictionaries	{{d:6914494}}	N/A
/OMNI-RESOURCES	{{d:6912864}}	N/A
/news	{{d:6908092}}	section
/includes	{{d:6908091}}	N/A
/happenings	{{d:6908090}}	section
/faculty	{{d:6908089}}	section

Figure 403. A sample canned report, illustrating directories in the system and associated dependency tags.

Development

Templating requires a developer. To create a “new page” template, you create a “.tmpl file,” which contains the HTML/XML code and a GIF file, which is a thumbnail of the template layout. Additionally, a Template Control File (TCF) is required as part of the new page template process and it defines the New Page Wizard UI that a user sees when clicking on a template thumbnail. A TCF controls one or more TMPL files by defining variables and customizing the form fields that users see when creating new pages. A TCF can also be used to echo certain values (either user- or administrator-specified values) into multiple TMPL files at once. For example, instead of creating a sub-directory and its supporting files and includes for that directory separately, a TCF can do it all at once. A TCF's syntax is XML based, and contains four structural tags:

- **<title>**: Defines the text that appears at the top of the New Page Template form that authors view when creating new web pages
- **<variable-list>**: Encapsulates a list of variable tags that can be used to present authors with one or more questions to answer when creating new pages, or administrators to set (hard code) values to be used by TMPL files
- **<template-list>**: Indicates which TMPL files to use
- **<navigation-list>**: Enables an automatic creation of navigational menu items for newly created pages

While this scripting approach may be convenient to some, it remains a proprietary approach. OU Campus is the only vendor using TMPL files. Also, TMPL may not be very attractive for your technical team to learn, compared to more popular alternatives, such as ASP.NET or any common Java framework. The actual look and feel is applied via XSL templates — something that your developers (or OmniUpdate) would need to code. While templating in general is not complicated, you will need to learn the OU Campus tagging library.

Publish Control Files (PCFs) are another paradigm in OU Campus. These are XML data files that specify how to process that data on publishing. In addition, your developers will use PCFs to indicate which bits of data on a page are allowed as editable regions in the CMS.

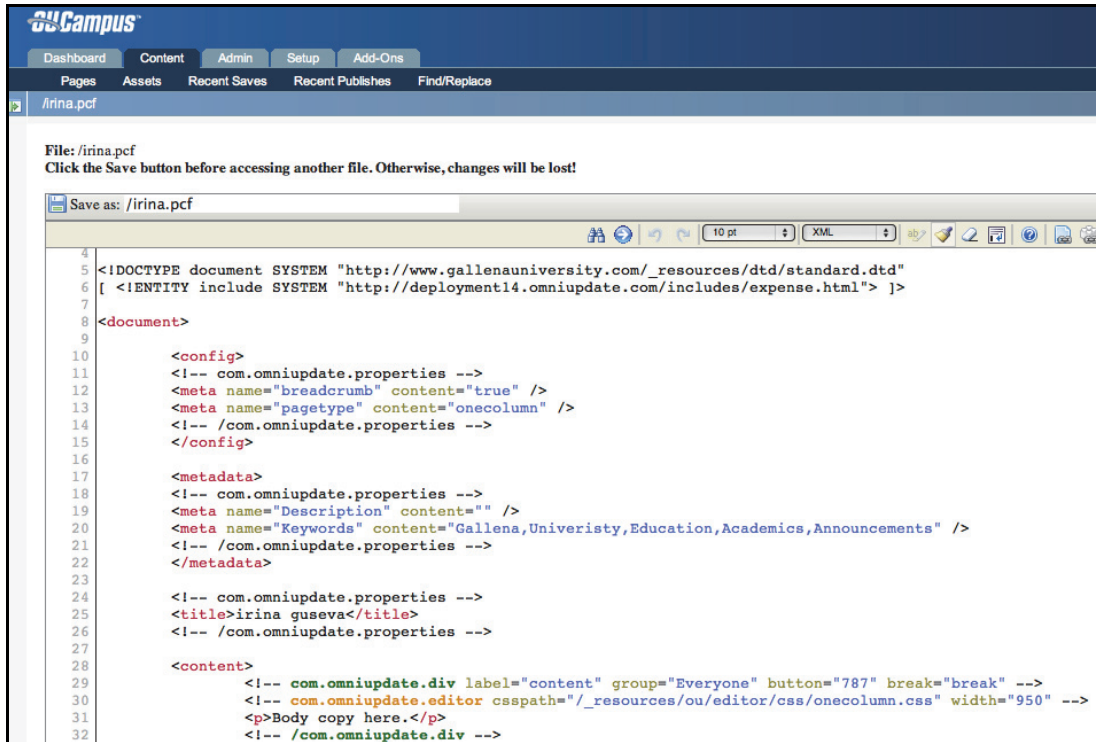


Figure 404. The OU Campus editable region tagging is a method for defining editable regions.

OU provides APIs that enable your developers to extend the application and integrate it with other external applications. Using these APIs, you can automate things like creating users, groups, pages, and even sites.

OU Campus also provides authorized developers with a web-based source editor to author code. However, make sure that you test it carefully if you are using Internet Explorer (IE) as your browser. One customer that we spoke to said that it didn't handle JavaScript optimally while using IE, thus leading to some performance issues. However, OmniUpdate claims that this was resolved with some tweaks to the source editor in v9.16.

Note that content types are fairly tightly bound to templates, and templates themselves are not as composable as you'll find in most other systems. This might work well for de-centralized university environments, but if your multisite management approach calls for more centralized control of diverse properties, this model will probably feel too clumsy and labor intensive.

In 2010, OU Campus migrated its query processor from Xalan XSLT to Saxon 9.3. This moved earlier versions from XLST 1.0 to 2.0, offering partial support for XSLT 3.0, XPath 3.0, and XQuery 3.0. OmniUpdate claims all customers are now utilizing XSLT 3.0, but it's worth noting that the 3.0 tools are still under "Working Draft" status and are not formally approved yet.

Performance

We have not heard of any particular performance shortcomings of the OU Campus management environment. Its internal benchmarking claims to be able to support 10,000

concurrent users and 1 million pages per instance, but each scenario is different when it comes to performance; make sure to test these parameters in your environment carefully. As always, publishing and FTP'ing large volumes of content will take time, and you may experience bottlenecks when you have a major editorial push.

Content

Contributor Experience

The company places a priority on simplicity over features, and this claim has been backed by customers with whom we have spoken. At least one customer feels that the company philosophy is, “If it will complicate the interface, we will not add.” At a time when many competitors are adding on to bulging super-user interfaces, simplicity is increasingly a good thing.

Other than a browser, content administrators do not need to download client-side software. OU Campus supports IE, Firefox, Safari, and Chrome in Windows, as well as major browsers in other operating systems (e.g., Mac and Linux). In higher-ed situations where operating systems often differ from department to department, customers laud the company’s commitment to browser compatibility in various OS environments.

Content can be in any language, including those that require double-byte character sets, although the interface is only available in US English. You can make the interface accessible via configuration settings. However, spell checking can be performed in a range of languages including English, Spanish, French, Italian, and Portuguese.

Contributing Content

For rich text editing, OU Campus offers TinyMCE — a well-regarded rich text editor, but you should test it extensively in your setting — especially if you copy and paste content regularly from other products, like Microsoft Word.

Contributors can employ a file browser to navigate to the page they would like to edit, or to create a new page. If you have multiple websites, they can all be managed under OU Campus, and upon login, an editor chooses the appropriate site. OmniUpdate does not offer as much easy content reuse across multiple sites as some other products within this report do. Managers should exert care here, because the product offers very little to help with ongoing management of multiple sites.

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

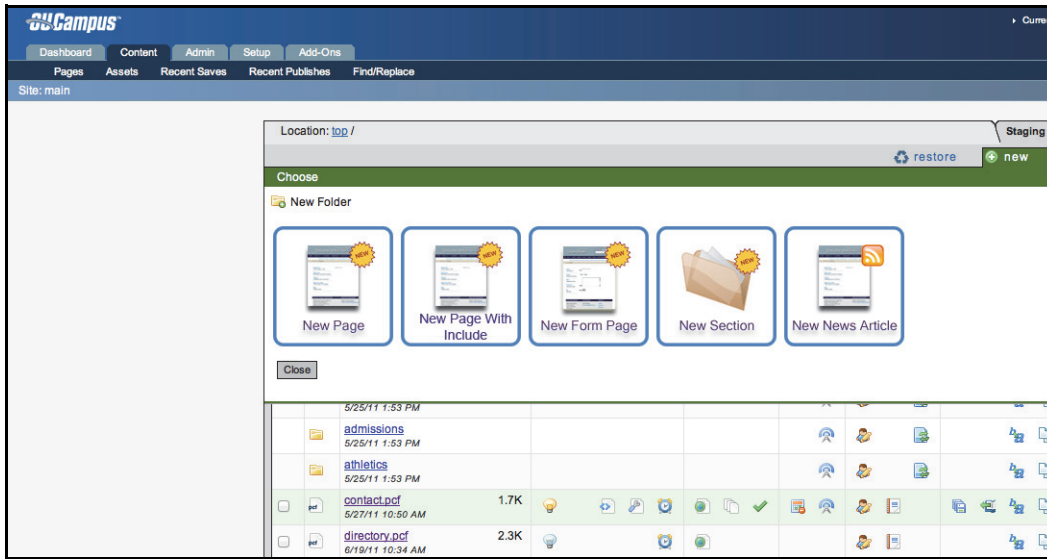


Figure 405. When creating a new page, you can choose from a predefined set of available templates.

In a similar vein, it’s important to remember that OU Campus is fundamentally a page-based management system. Do not consider it for more complex, componentized content management. To be fair, OU Campus Web CMS offers a feature called “Snippets,” which is a pre-formatted HTML fragment that users select and insert by clicking a button while using the WYSIWYG editor. You’d only want to use this for very commonly reused elements.

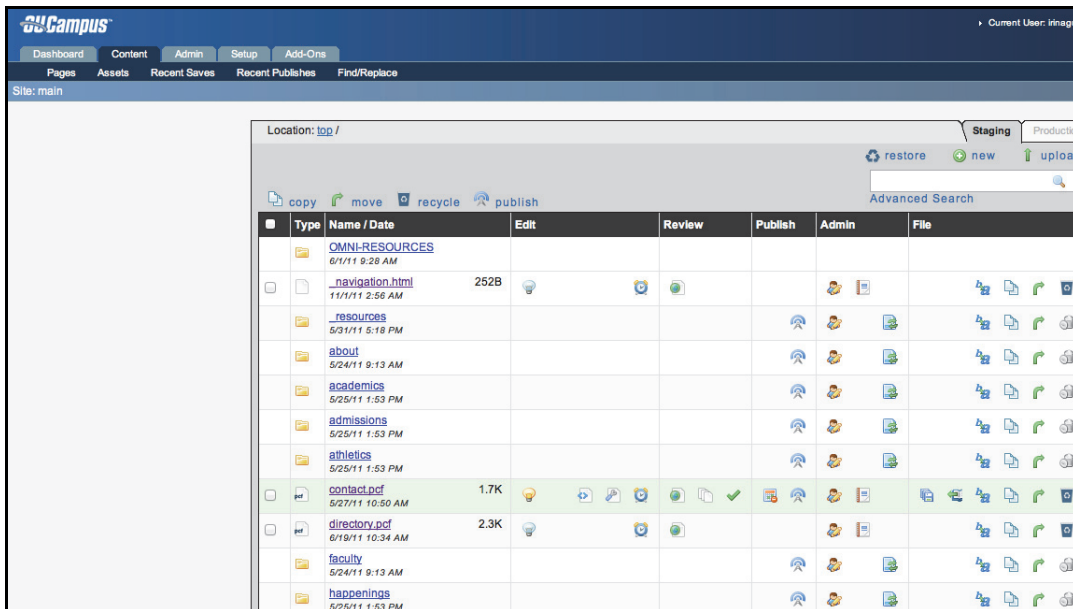


Figure 406. A view into the folder and file structure in a sample implementation of OU Campus. You really need to know the structure of your site to be able to navigate this browsing experience.

Another potential option for content reuse is Asset Manager, one of the latest additions to the CMS. It somewhat mimics what is already possible with Snippets, but with Asset Manager, you can reuse elements — such as words, phrases, and paragraphs — within the WYSIWYG editor. Asset Manager works with the following types of assets:

- Web content: text, images, links, etc.
- Plain text that follows formatting of the page template
- Source code
- Live delivery, platform-based image galleries
- Forms, polls, and surveys

Some OU Campus customers appreciate the granularity of reusing words (below), but for others it is too much overhead. The most common use cases for Asset Manager include simple reuse of images, media, links, and code assets across multiple sites. This is managed via custom security groups that define access (e.g., read and edit, read only), and your security model requires careful planning to ensure that you don't end up with either over-restrictive or clashing profiles.

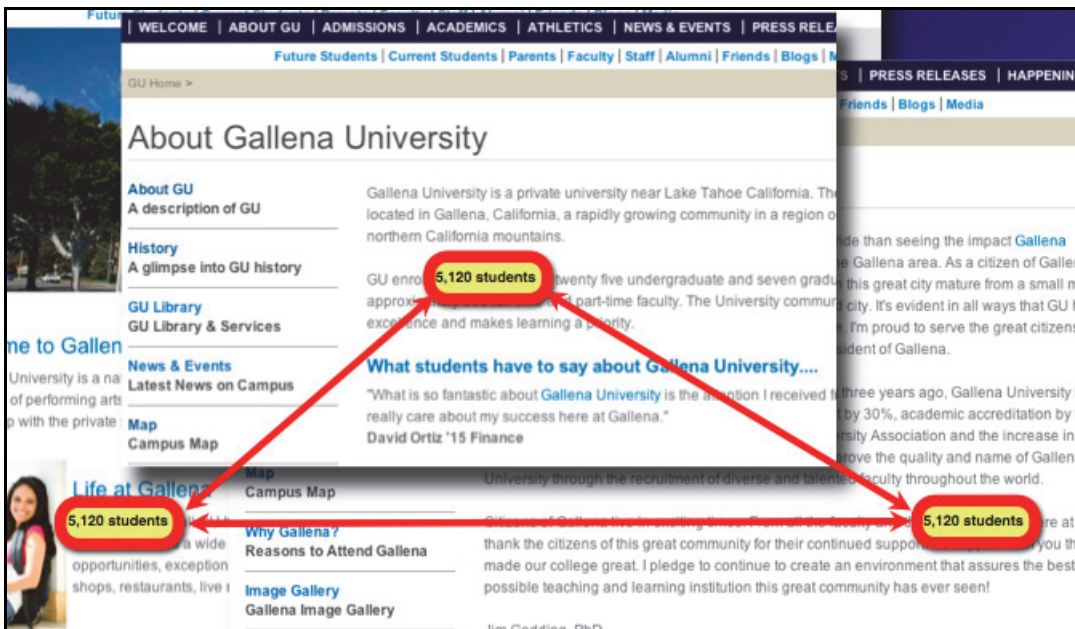


Figure 407. An example of phrase reuse using OmniUpdate's Asset Manager.

OU Campus supports in-context editing, (again based on permissioning), which shows what you can change and what you cannot. This is called “JustEdit.” OmniUpdate demos sometimes even show dynamic content edited in this interface. Note however, that the management environment will only “virtualize” a PHP application; if your delivery environment runs on .NET, for example, you’ll need to push content to staging to preview it properly, or use custom scripting to display content from production.

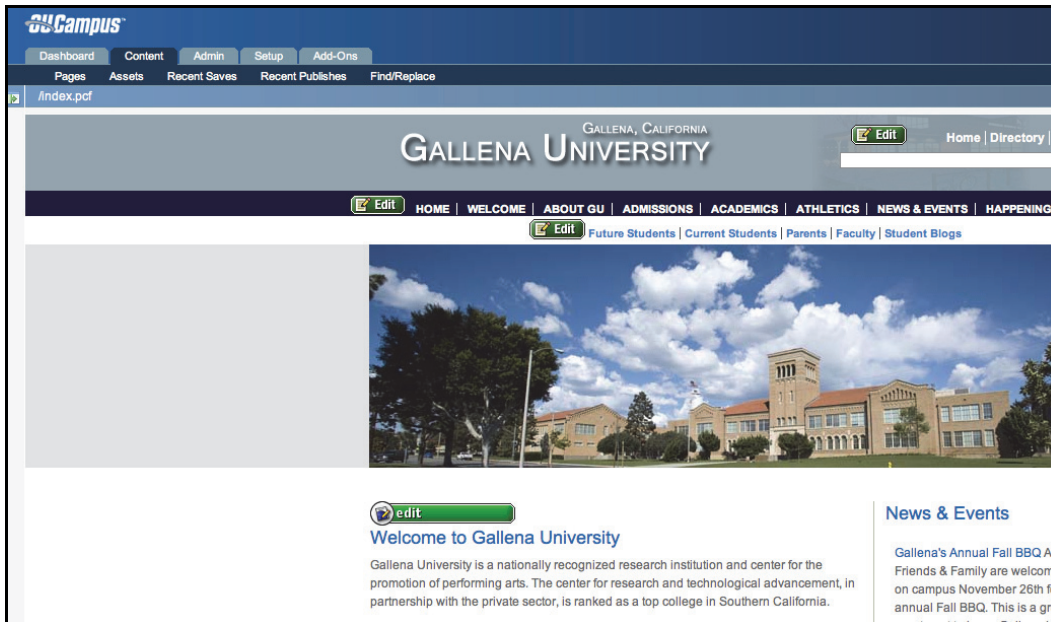


Figure 408. Green “Edit” buttons indicate editable areas in the JustEdit interface.

There’s a peculiar dimension to JustEdit. When visiting your own site, you may find a page that needs to change. Instead of the usual approach of logging into the CMS and finding the page, OU Campus offers what they call “DirectEdit Link.” Typically, the last date the page was updated displays in the footer of each page. This serves as a link, and takes you (via login), directly to the editorial interface where you can update the page.

This is certainly useful, but it’s also a good example of how ease of use can introduce a security liability by providing access in this rather overexposed way. Make sure to consider the risks it could introduce. You can let it remain as a default, customize its positioning, or remove it. We recommend that you remove the login link from your web pages. The company praises its security, including the fact that login pages are SSL encrypted. However, we’ll note that although nearly every other vendor in this report could work this way, none of them do.

OU Campus offers nice preview functionality if you need to support multiple browsers. Using the Multi-Browser Preview (Figure 409), you can run a script that generates images of what your page will look like in a number of different browser and operating system combinations.

If the script generates snapshot images that indicate any functional page errors, you can identify major formatting errors. Certainly a handy feature, be advised that it does not replace proper browser testing.

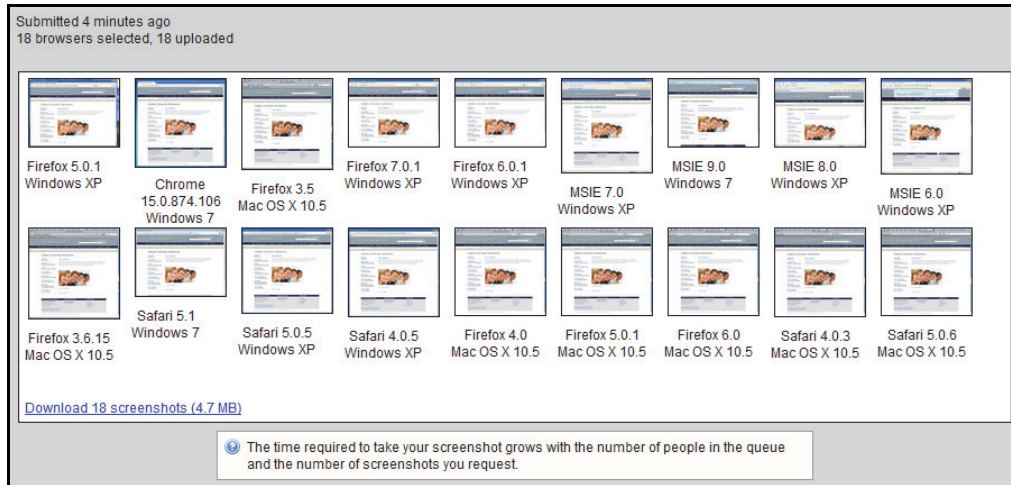


Figure 409. Screenshot generation may take a bit of time, depending on how many browser previews you want to generate.

To improve accessibility (and Section 508 / WCAG compliance), OU Campus has automated prompts, which alert editors and template developers when their content or code breaks accessibility. Site Checker functionality includes the ability to check link health and identify broken links, as well as set rules for redirects and time-outs. Currently, reporting on overall site health in this regard is limited. Additionally, the Final Check function allows a user to administer some tests — including spelling, link checking, and W3C compliance — to validate whether or not a page is ready to publish.

OU Campus also offers a multi-item find-and-replace feature. Available only to administrators, you can scope changes to individual folders, and the feature helpfully maintains an “undo” step, as well.

The OU Campus CMS is exceptionally weak at asset management. OU Campus wrote its own image-editing widget (a Java applet — test this carefully) that you can use to resize, crop, optimize, and rotate images. However, the repository offers no metadata for images or other binary files. It simply maintains size and modification dates, and it is not extensible. Similarly, although it provides “Transcode-It” so that you can convert all your videos from one format to another, it doesn’t provide any features specifically for managing digital assets.

Zip Import *might* be helpful here to compensate for OU Campus’ long history of suffering from the lack of a usable way to manage batches of assets in a pain-free manner. Not that this has become completely pain-free now, but at least content contributors can utilize the classic notion of zipping a bunch of files together and dropping them into OU Campus CMS.

Figure 410. Zip Import might be helpful here to compensate for OU Campus' long history of suffering from the lack of a usable way to manage batches of assets in a pain-free manner.

Of course, all .zip files would need to be extracted (and the CMS does auto extraction in most cases) and display folders in expanded view upon upload, allowing users to modify files (rename, delete, etc.) before the extraction commences. Some OU Campus customers note that while this feature may not be as annoying as uploading files one by one and clicking ten browse buttons, it is still far from efficient and user friendly.

Content Lifecycle

Basic approval workflows are included in OU Campus Web CMS. Workflow actions trigger automatic email to the next person in the process. The workflow email comes with three direct links:

- To approve or decline the page
- To the editor's OU Campus inbox
- To view the original page

Workflow is set up for basic approvals only, with few fancy bells and whistles. For example, you can only send your page for approval to an individual, not to a group. However, it will support elaborate or long approval chains, as you might see in some high-ed environments.

OU Campus Web CMS also sports a "stale-page reminder" feature. Users can send notifications for individual content owners or groups of users when pages exceed a specified time frame without being edited.

The platform also offers a "recycle bin." It's not a real archival system, but it allows authorized admins the ability to restore selected deleted items — either singly or in groups.

Experience

Publishing

As a decoupled, file-oriented system, your URLs are what you make of them, and you have to manually specify each URL as you create a page.

Dependency Manager is a feature that allows you to monitor hyperlink integrity in the CMS. In case a page gets moved or renamed, the manager updates the URL in the referenced pages.

You can publish an automated RSS feed, and multiple feeds can be aggregated into one single feed to which users can subscribe. This grouping is especially handy in higher-ed environments where RSS subscribers will want to be notified of new content from multiple content authors, such as all content from professors in a certain department. A few licensees report that it is not easy to update the feed when editors change existing content (revisions to existing content must be syndicated manually). Additionally, the “sitemap.xml” (which contains the SEO “must-have” site structure map) is automatically updated during the publishing process; this is often overlooked by some other vendors.

Digital Marketing

There is no native analytics functionality in OU Campus, but the product comes with an “Add-ons area,” where you can link to external services, like Google Analytics.

If you plan to launch a blog, OU Campus has a module called OU Blogs. It enables integration with WordPress (a popular open source blogging platform), and supports XML-RPC compliant blogging systems. You can create content using the OU Campus interface, which is then published to WordPress. It also connects to the workflow system if you want new postings to follow an approval process.

OU Social module is targeted toward those dabbling in social media efforts, such as Facebook pages for colleges and universities. While many higher-ed institutions are quite active across the landscape of social media technologies, the strategy behind this effort is often underdeveloped. While OU Campus CMS will not solve the issue, it does provide a more automatic way of keeping a college’s Facebook page updated. What happens is that you, a college or university, start a Facebook page, OU Campus then installs its own Facebook application (written using Facebook APIs) that communicates with the said college Facebook account and allows for content to flow from the CMS into the Facebook page via communication with OU Campus’ own APIs.

Experience	
Publishing	
Standards Adherence	<input checked="" type="checkbox"/>
Multichannel	<input checked="" type="checkbox"/>
Mobile	<input checked="" type="checkbox"/>
Digital Marketing	
Site & Campaign Analytics	<input checked="" type="checkbox"/>
Testing & Optimization	<input type="checkbox"/>
Segmentation & Personalization	<input checked="" type="checkbox"/>
Social Media Integration	<input checked="" type="checkbox"/>
Promotional Campaigns	<input type="checkbox"/>
Community & UGC	<input checked="" type="checkbox"/>
Workplace	
Collaboration & Networking	<input checked="" type="checkbox"/>
Dashboard	<input checked="" type="checkbox"/>
Ancillary	
Site Search	<input checked="" type="checkbox"/>
Online Forms	<input checked="" type="checkbox"/>
Module Ecosystem	<input checked="" type="checkbox"/>

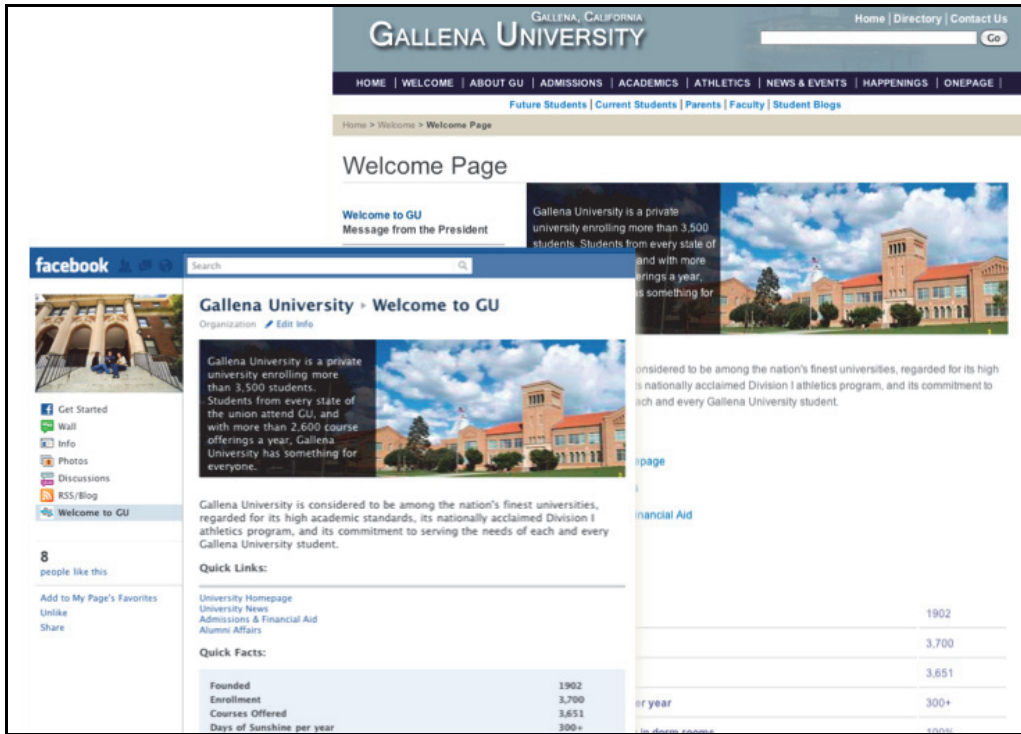


Figure 411. Your Facebook page can reuse the same content as your website home page.

On the Twitter side of the universe, OU Campus has introduced the bit.ly URL shortening service, with the ability to send posts containing short URLs to Twitter or Facebook from the CMS UI at publish time or when scheduled. However at this stage, bit.ly is the only supported URL shortening service, which doesn't provide much flexibility if you're compelled to explore the 100+ other services, including is.gd, 301url, TinyURL,youtu.be, or goo.gl.

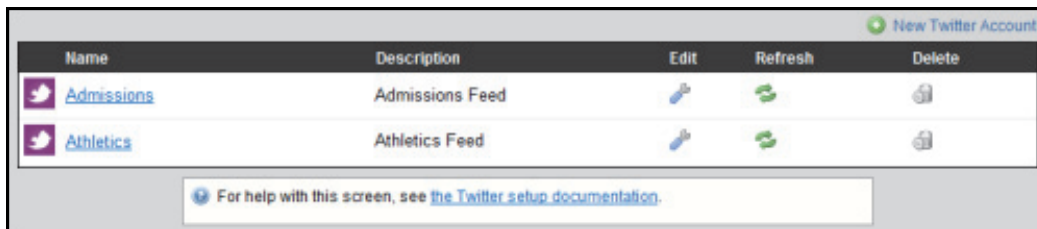


Figure 412. A view of sample Twitter accounts set up in OU Campus.

As of version 9.11, OU Campus introduced the server-side module of the Live Delivery Platform (LDP). While it has nothing to do with the delivery tier, it provides you with the ability to create and manage forms, polls, image galleries, and surveys. With the LDP module, you can build forms (including text, radio buttons, checkboxes, etc.) and drag and drop fields to reorder.

If you embark the mobile channel wagon, your out-of-the-box options with OU Campus will be limited. There's nothing in the product that will provide mobile delivery capabilities. However, OU Campus found a creative solution to this problem — Professional Services.

They will offer you a special consulting package (at an additional cost) to help you deploy a mobile site in HTML5 and jQuery mobile framework.

Ancillary Services

In theory, you can provide more run-time interactivity via plugins. However, some developers reported problems extending the system by adding plugin components (such as calendaring applications) without the help of OmniUpdate support.

One of the important developments by OmniUpdate is the introduction of Quick Search in version 9.17.

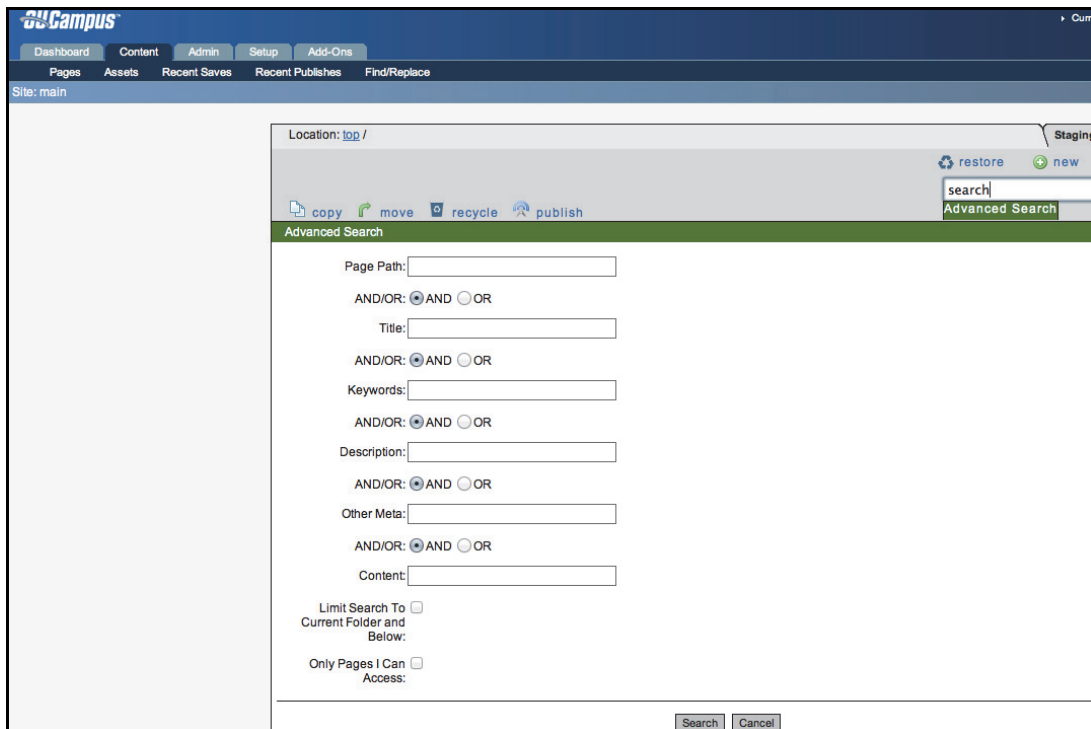


Figure 413. Advanced search is available in addition to simple, keyword-based search.

While the original search product called OU Search still exists in the CMS and is designed to provide front-end search capabilities, Quick Search is a separate and independent feature also based on the Lucene search engine that performs the same task within the back office. The curious thing about this search implementation is that prior to its introduction, OU Campus CMS users would navigate the directory tree, but go to published site itself to find the actual content.

Vendor Intangibles

OmniUpdate is a privately held company, and remains closed about its financials. According to the company, they are profitable and 95 percent of the revenue comes from software. As is typical with SaaS products, the license model is subscription based. Typical deals start at US\$21,000/year and go up to \$70K annually. The company also reports that in recent years they saw an increase in average deal size. What this means to you is that your first year cost can run up to \$65K. If you're interested in any of the third-party modules like campus maps or tuition calculator, add at least \$2,500 – \$5,000 per year to your bill.

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

While some projects are implemented with help from OmniUpdate, you can also work with their business partners, which include some boutique web agencies and smaller studios.

The company is averaging a major update yearly, with more minor releases nearly every 3 months. Customers lauded the company for its knowledge of higher education: how to deal with distributed departments, higher-ed budgets, and generally non-technical users. Customers typically regard support as very responsive and praised the company for being driven by user requirements. Recently however, customers feel like the relatively small company is experiencing difficulty in keeping up with client growth

Some say that the resolution of support requests is lagging more than in past years. The company says it has responded by doubling its support staff (which now constitutes around 40 percent of the staff), claiming that it has reduced resolution times by half.

You can find a publicly available support site (support.omniupdate.com), offering technical documentation for developers, and there are training videos available for end users. OmniUpdate also arranges annual user-group meetings in California, and hold a client workshop series focused on specific subjects every fall. OmniUpdate customer attendees find even more value than most, due to the homogeneous nature of the customer base. OmniUpdate has also set up OmniUpdate Community Network (OCN), which is a customer community site (based on Ning) at ocn.omniupdate.com. The goal here is to provide a place for about 20,000 members to share tips, code snippets, and best practices, as well as vote for upcoming product features.

Conclusion

OmniUpdate certainly has many references in higher education, where its SaaS model seems to go over well. We find the product a potential fit for simpler and/or highly distributed websites, where a page-centric model and sparse user interfaces work well.

While the company has experienced growth in recent years, it is still comparatively small. Its partner base is also limited, and it can be difficult to find experienced help. This may not be a problem for you, but in a crowded and competitive market, make sure to assess your risk profile carefully.

Like Ektron, OmniUpdate is an ambitious company that seems more interested in generating new features than testing or elaborating them carefully. Even if you run a university site, make sure that this is a good cultural “fit” for your team before signing the dotted line.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

OpenCms: OpenCms

opencms.org

Project at a Glance

Specsheet	OpenCms 9.0.1 Summary
Geography	Global, with emphasis in Europe, particularly Germany
What's New	<ul style="list-style-type: none"> • Improvements to page authoring and in-context editing
Strengths	<ul style="list-style-type: none"> • Core source code and binaries are truly free • Strong repository services and excellent search • A decent ecosystem has evolved, particularly in the DACH region • Product development proceeds at a steady, reasonable pace • Bugs are exposed in an openly-available bug tracker • You can purchase optional modules for enterprise functionality (e.g., replication, LDAP) • Repository supports CMIS
Weaknesses	<ul style="list-style-type: none"> • Quality testing is done partly by Alkacon Software GmbH, partly by the community, and partly by customers already in production • Dearth of asset management services makes it a poor fit for media-rich sites • Default interface is geared toward power users • Almost complete lack of digital marketing services rules out many modern web publishing scenarios • Clustering and replication require additional commercial modules • Add-on modules can be difficult to install and manage as there is no formal third-party module validation or testing process • Commercial support options are limited • Alkacon itself competes with erstwhile consulting partners, which may limit growth of the community
Potential Fit	Informational Site, SMB/Departmental Digital Workplace
Unlikely Fit	Community-Oriented Site, Basic Digital Marketing, Complex Scenarios
Compare To	Plone, Hippo, Magnolia
Operating Systems	Windows, Linux (Red Hat, Debian, SUSE)
Repository	Databases: Any JDBC-capable database
Client	Browser: IE, Firefox, Safari, Chrome
App Platform	Java (6 or 7). Prefers Tomcat but runs on multiple servlet containers
Licensing	Core: Open source — LGPL, plus optional commercial modules
Ownership	Alkacon Software — GmbH owns most copyrights

Summary

What sets OpenCms apart from the other open source platforms is its longevity. Led by project founder Alkacon Software GmbH, the community has slowly evolved into what is now a decent ecosystem of customers, consultants, and third-party add-on providers — albeit focused on the DACH region. As one of the first Java-based open source WCM systems built atop dozens of well-known, open source components (with name WCM tools), the best thing we can say about OpenCms is that it has endured, if in a somewhat musty way.

The platform has very good user management, and is customized using good-old JSP. You'll find a decent community around OpenCms, with an annual conference in Germany.

As with other open source content management systems, the OpenCms user interface wins no accolades for usability, and certain key features are either poorly implemented or absent, altogether.

Consider OpenCms for simple business scenarios, in particular basic interactive projects or information-rich static sites, where OpenCms could get you up and running with a decent solution without an excessively long implementation time. Do not consider OpenCms for anything more elaborate unless you have substantial in-house Java expertise, and are willing to bear the burden of building and testing your own features.

Also, if you need various “enterprise” features (such as proper staging environments or LDAP connectivity) that the platform natively lacks, then you either need to build it yourself (not a promising idea), or buy proprietary commercial modules from project sponsor, Alkacon.

Introduction

OpenCms has been in active development since 1999. It was first released into open source in 2000 by a German new-media agency that had built the tool originally for use by its clients. The “OpenCms Group” was a team of individuals who initially coordinated the OpenCms project, and some of them went on to found Alkacon Software, which is the dominant commercial player for the platform. Older parts of the OpenCms source code still have the name “OpenCms Group” attached because of copyright reasons.


Initially, the project attracted interest — particularly in Germany and elsewhere in Europe — because it was the only Java-based open source Web CMS tool of any repute available. In the meantime, other Java-based, open-source content management systems have gained in popularity, including Hippo and Magnolia.

Alkacon is a private consulting firm based in Cologne, Germany. In addition to organizing the community, it provides support, training, and — critically — implementation consulting. Alkacon also sells commercial add-ons under the name “OCEE” (OpenCms Enterprise

Scenario Fits

Simpler Site

Informational 

Microsites & Landing 

Mid-Range

Basic Digital Marketing 

Mobile Site 

Community Oriented 

Complex Site

Advanced Marketing Portal 


Global Enterprise 

Multichannel Publishing 

Ultra-Large Single 

Digital Workplace

SMB/Departmental 

Global/Enterprise 

Extensions), which provide more enterprise-y features like LDAP connectivity, clustering, replication, and caching services.

Version 9 was released in 2013, and it was a major upgrade with significant new functionality, most notably around multisite management and drag-and-drop editorial functionality. Since then, Alkacon produced a minor bug-fix release (9.0.1) in February 2014.

Technology

Technical Administration and Security

OpenCms is written in Java and runs on Tomcat by default, but is capable of running inside most modern servlet containers.

The CMS comes inside a single WAR file that once deployed, requires access to a JDBC-compatible database. Your most straightforward install will run the default Tomcat and MySQL combination.

In general, you'll want to be careful about straying very far off the beaten path (in terms of known configurations), because while Alkacon does unit testing on many pieces of core code, the QA process for OpenCms (as with most open-source products) rests partly upon customers. That's particularly true for non-core features contributed by the community.

OpenCms stores everything in a Virtual File System (VFS), which is actually part of the database. The package offers shell access to the OpenCms database, as well as developer APIs and a web-based, "Explorer" interface. The repository is CMIS compliant, so other systems can access it using that standard. XML-based configuration files round out the platform.

Because of the way OpenCms mirrors JSPs in its virtual file system, OpenCms needs to intercept page requests at run time in order to pre-process certain directives inside JSP tags. This pre-parsing technique results in a number of nonstandard behaviors. For example, the <jsp:forward> tag simply does not work in OpenCms. Since OpenCms is an extraordinarily JSP-centric system, you'll want to research the limitations of this aspect of the product very carefully before you put it on your short list.

As with many other open source alternatives, OpenCms relies on a concept of modules for packaged extensions. A module usually consists of a set of templates, images, Java classes or libraries, and other resources. Using the browser-based administration UI, you can manage, create, import, export, and delete modules. Normally the .zip format is used for modules, reminiscent of Eclipse plugins. Compared to the relatively easy installation of the core package, maintaining the extra modules, installing them, and testing them — potentially across multiple servers — can be a time-consuming task. Note that for each new major release, all modules generally need to be rebuilt.

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

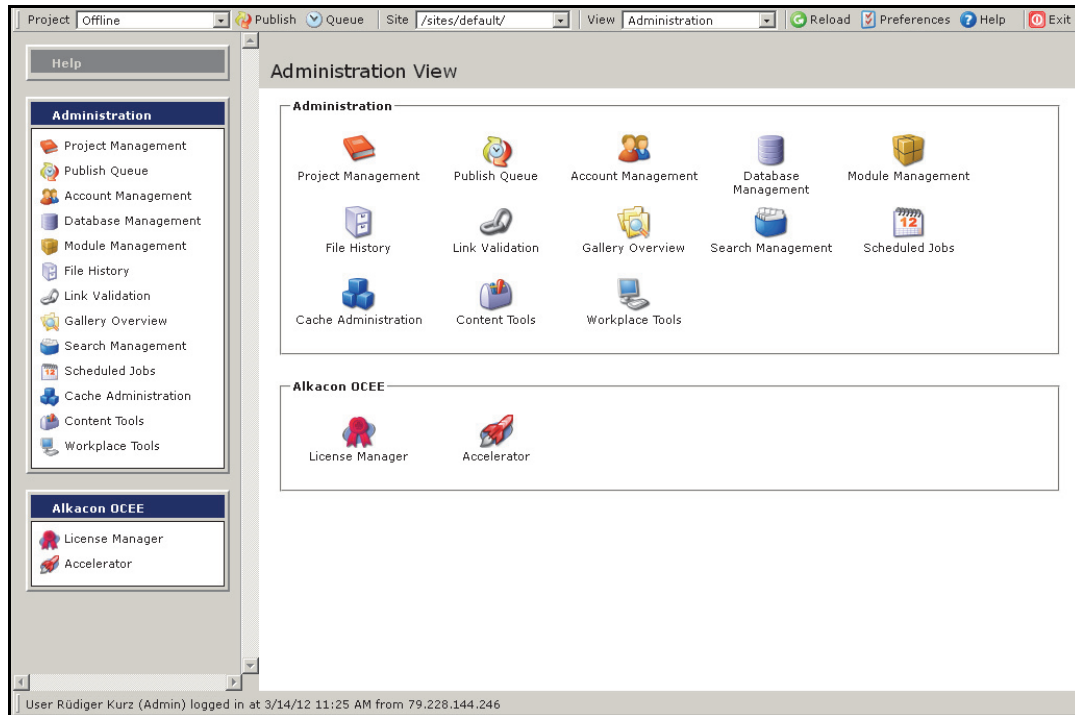


Figure 414. The OpenCms administration console can do the basics in a browser, but may soon turn to configuration files and shell commands. Source: OpenCms.

Although it's possible to create a fully decoupled system using OpenCms (more on this later), by default you're working in a fully coupled system wherein a single instance of Tomcat hosts both your live web pages and the OpenCms WAR. OpenCms does have a notion of "Offline" and "Online" projects, which provides a sort of emulation of staging and live areas. Under this approach, you change all content and code in an Offline project, and then publish them to the Online version. However, both Offline and Online run off the same instance of OpenCms — there's no support (in the free version) for multi-server topologies. In an enterprise scenario, this might not be acceptable, because typically staging and live environments have different requirements in terms of usage, caching, delivery, security, and so forth.

If you want proper separation of concerns, you can acquire it commercially via OpenCms Enterprise Extensions from Alkacon. One of the extensions provides replication capability; another provides clustering. With OCEE Replicator, you can separate database instance on separate machines — e.g., a staging server inside the firewall, and a delivery server in a DMZ — enabling a decoupled topology.

OpenCms offers both static and dynamic publishing. Publishing can be started manually or automatically, using time-based content release or expiration. If you choose to run everything on a single server machine, OpenCms can handle promotion of content simply by marking items as either "Online" or "Offline."

Unlike adopters of other systems that are also capable of both static and dynamic delivery, many OpenCms users publish to static files. For content that is not often updated, this can be a cheap and attractive option. The default configuration places the files in a folder structure called /opencms/opencms/website. This provides clean (if lengthy) URLs, and is found at

most OpenCms-powered websites today. Depending on your requirements, you can schedule your publishing at given intervals (e.g., daily), or invoke a manual publish.

OpenCms has a built-in user and permission management system (Figure 419). An administrator defines users and groups using the Account Management console in the Administration View. This enables you to control who can access which resource. Unfortunately, LDAP integration (for management of users and groups) is not a core feature of the product, although Alkacon does offer a commercial LDAP connector as part of its OpenCms Enterprise Extensions package. (The package costs €2,500 to €6,500, depending on the level of support desired).

The core platform has an unusually good user management. There are additional roles, support for organizational units (LDAP-style OUs) which can be mapped to different sub-trees in the OpenCms content hierarchy, and the ability to batch-import users using a CSV file. There is also a notion of a “sub-administrator,” where a user can exert administrative control scoped to an OU, which is a handy feature if you need delegated administration.

Access permissions can be set individually on each content asset in the system. Folder-based rights are inherited, unless otherwise overridden.

Apart from log-file generation, OpenCms does not offer much in the way of managerial reporting.

The overall administrative interface is browser based. A unique feature is “CmsShell,” which is a command-line interface to access all resources in the OpenCms. Depending on your orientation and skills, you may consider this an unwelcome, “techie” feature, or a powerful short-cut mechanism.

Development

In theory, business users can create new content types in a browser. In practice however, the process is quite a bit arcane and ultimately requires a developer. A developer needs to create an XML file (in fact, an XML Schema) with details of fields required for the specific content type. OpenCms then automatically generates user-friendly forms for business users to create content in, using a WYSIWYG editor. Any developer with a modicum of XML and/or JSP skills can create custom content types or extend the page templates quite readily (it’s not like you need to hire an OpenCms ninja, per se), but the process will prove opaque, at best, for non-developers.

Templates are written in JSP with taglibs to access common OpenCms functions. Tag library functionality also is accessible using a scriptlets API. Templates are managed within the OpenCms repository, and hence can be versioned, packaged in a module, or made a part of a workflow.

The templating system is a somewhat homely admixture of XML, XSD, JSP, HTML framesets, and JavaScript. To be sure, it’s all standards-based stuff that any competent Java developer immediately will feel comfortable using — although a non-developer will get immediately lost.

Significantly — and unlike more modern WCM tools — OpenCms commingles logic, layout, and slotting, all in one file. It’s simpler, but less flexible for more dynamic publishing organizations.

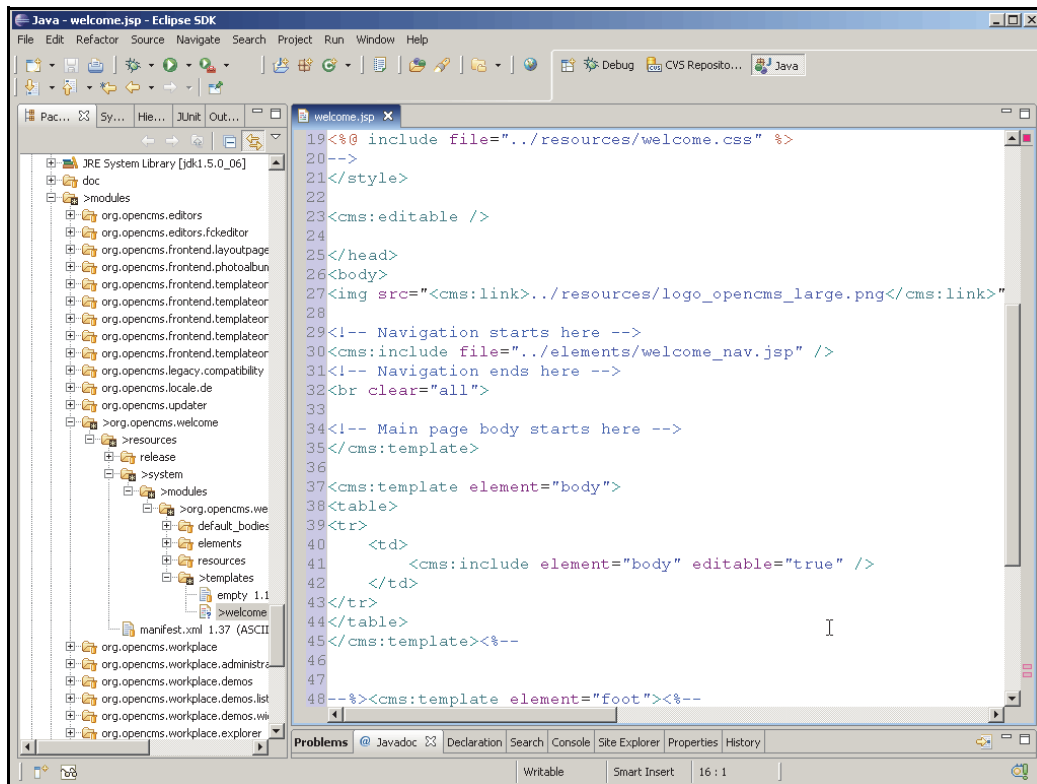


Figure 415. Building and customizing templates in OpenCms.

An elaborate sample site ships with the basic package. It can be used as a basis of your own site, but beware that new versions laid down during version upgrades will overwrite these.

OpenCms supplies a browser-based editor in which you can write JSP code if you don't feel like firing up Eclipse. Most developers likely will want to work offline and employ the WebDAV service to synchronize code.

Performance

Pages with dynamically built content can quickly bog the system down due to the sheer volume of requests to the database. OpenCms has a so-called FlexCache mechanism for allowing appropriately tagged page elements to be cached, but it is currently used only to cache JSP and XSLT, rather than all resource types. The FlexCache is an improvement of the "Element Cache" of earlier OpenCms versions, but its use still requires a lot of fussing with JSP internals; it's far from a push-button affair.

If you're serious about maximizing the performance of the system, you'll almost certainly want to take advantage of the OCEE Accelerator extension that comes with the (added-cost) OpenCms Enterprise Extensions package sold by Alkacon.

Note that there is no built-in support (in the freely available product) for multi-server environments — that is, separate staging and production servers, or clustered delivery environments. Proprietary modules for clustering and replication are available from Alkacon for €2,500 to €6,500, depending on the level of support desired. However, the developer

community, for whatever reason, doesn't seem active in providing this kind of "enterprise-grade" functionality for OpenCms.

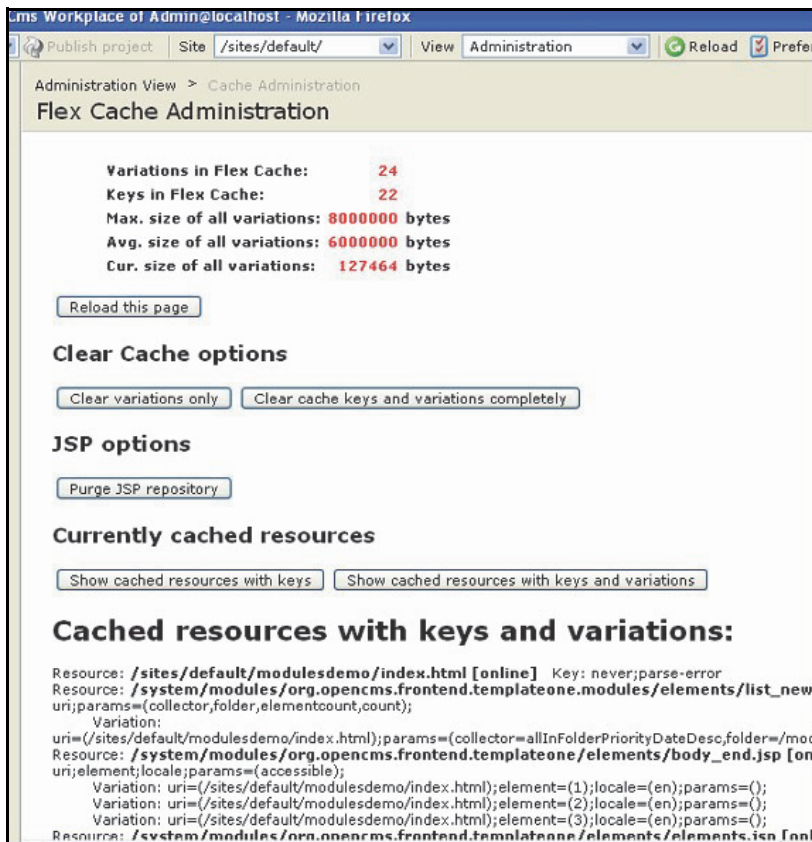


Figure 416. Flex cache provides the ability to define page variations using keys and improves the runtime performance of the site. As you can see, it is best managed by a trained administrator.

In terms of scalability, one customer claims to run a hundred sites (with pages numbering in the hundreds of thousands) off a single instance of OpenCms, with good delivery-side performance. That doesn't necessarily mean you'll be able to get the kind of performance you need in your particular situation (only testing can determine that), but it's encouraging to know that the potential does exist to manage large sites efficiently, which is not true of all open-source content management systems. (Compared to Joomla!, for example.)

Content

Contributor Experience

The look and feel of the OpenCms UI is somewhat staid and dated. In terms of overall usability, the product is best judged as “fair,” and it can certainly hold its own with the likes of Plone.

Interfaces ship out of the box in both German and English. Community-built localization kits are available for French, Italian, Slovak, Spanish, and Brazilian Portuguese, but these are all dated and most do not work with the latest version. Remember, these are community-provided modules, they come with no warranties, and sometimes little quality testing.

If you need to translate the UI and Help into other languages yourself, there’s a special localization tool designed to facilitate editing and management of the appropriate resource bundles. Alkacon also has (free) documentation for this process.

Contributing Content

Like most tools, you’ll see two editorial UIs:

- An in-context editorial experience
- A “back-end” repository explorer, called Workplace

Both UIs will probably be a good fit for the trained and experienced power user. This is something you’ll want to test for yourself.

The in-context experience allows you to navigate the site and edit individual items in-place with an HTML5-based editor (handy), as well as drag and drop new content items using a content-picker pop-up. The picker is feature-rich, but not particularly intuitive, and you will need to rely on search in a large repository. You can do things like define navigation and sub-navigation structures using the picker.

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

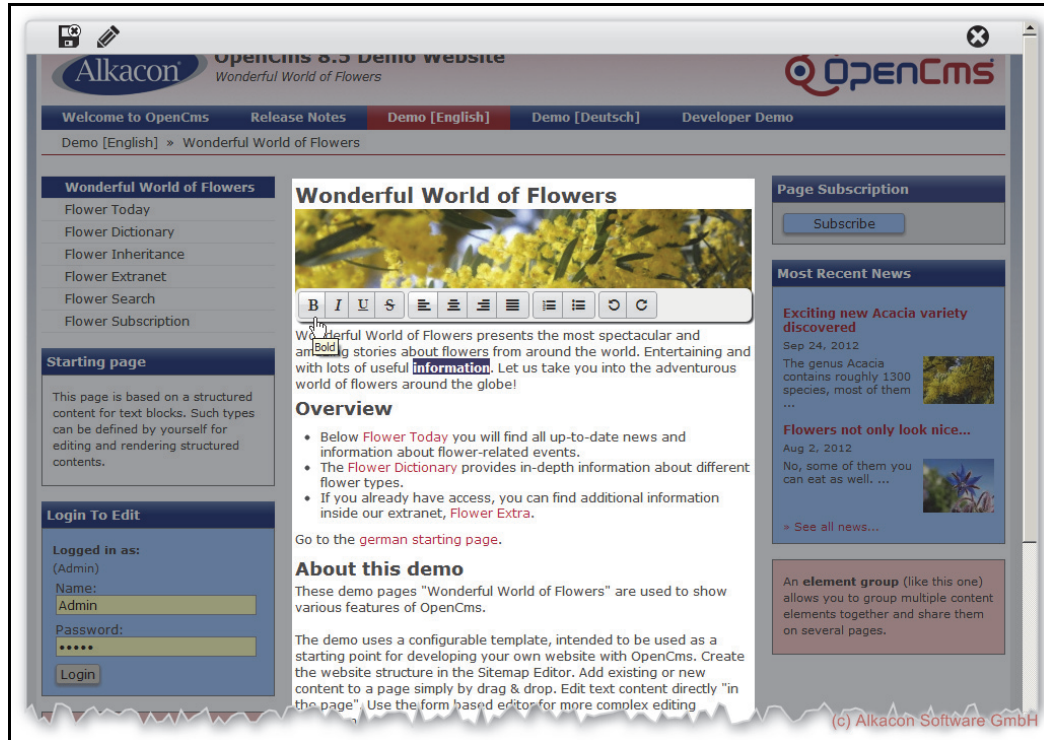


Figure 417. Inline editing. Source: OpenCms.

OpenCms uses the TinyMCE editor, having switched a while back from the CKEditor. The separate (and older) Workplace UI is a tree-based navigation tool, and allows you to manipulate content items in a more traditional way.

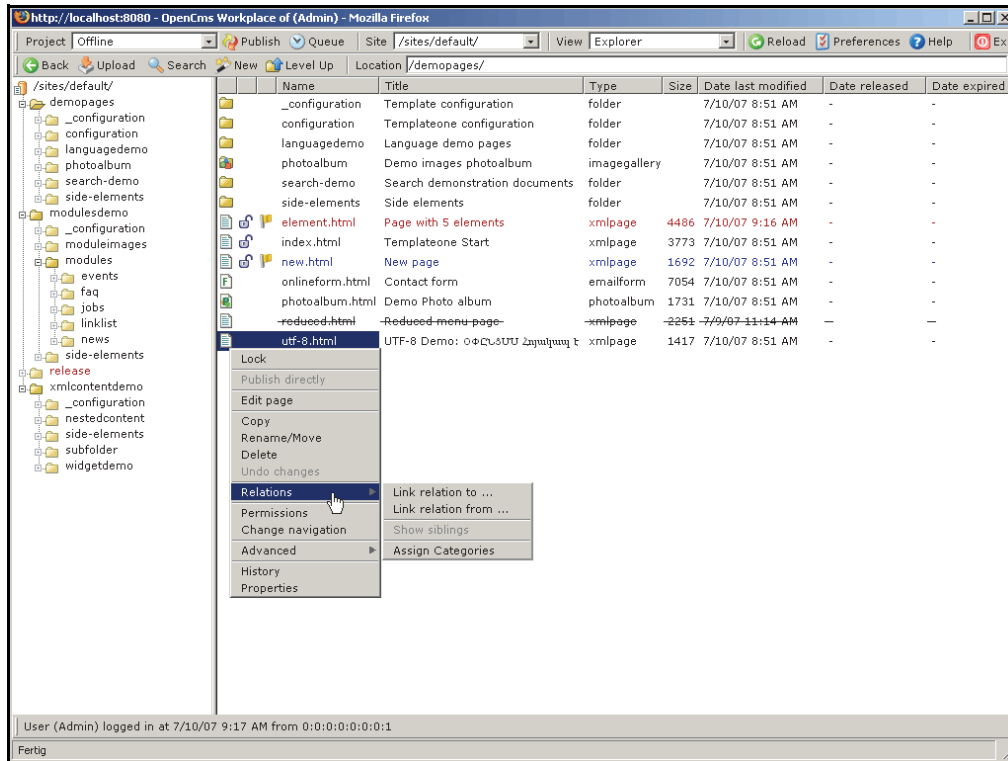


Figure 418. The Workplace view gives you a powerful — if somewhat prosaic — set of editorial options. It's more for the manager than the author. Source: OpenCms.

The package includes a nice internal link checker. This is integrated into publishing, so the publishing process will alert — and even pause — if link problems are identified. The UI also has tools for inspecting linked-to and linked-from properties of individual assets.

OpenCms offers a feature called “Content Relationship Engine” (CRE), which allows you to manage internal linkages, as well as link different resources (files and folders) using arbitrary relationships — essentially collections of content. This ensures that when you publish a resource, all related resources are also published, and it enables you to publish content packages — like site-wide promotions — all at once.

This is a very useful feature if you have content in multiple languages and you want to ensure that when you publish an article in one language, the same article in other languages also will be published. You also can tag content with categories and then display all matching content for a particular category using the APIs.

The content collections motif does not work very well for images. You can gather images into carousels and such, but not in a particularly visual way. OpenCms offers the standard browser-based image editing functions, but otherwise it is unusually light in terms of asset management.

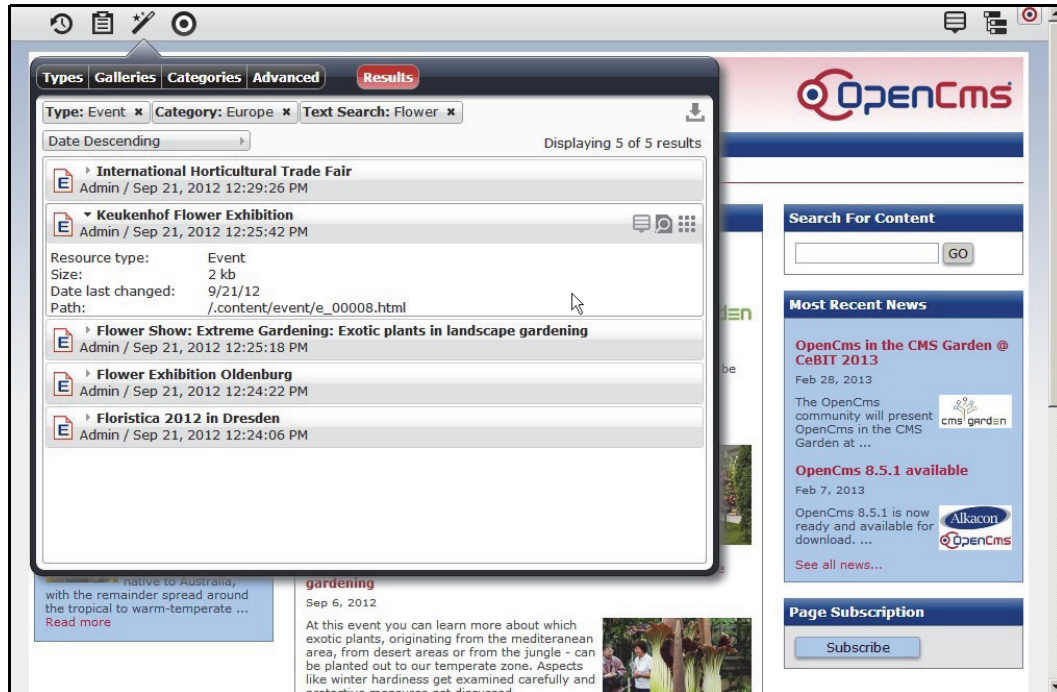


Figure 419. Putting together an image gallery is possible, but not particularly intuitive.
 Source: OpenCms.

Content Lifecycle

With OpenCms’ “project management” feature, you can group files and resources together, and then collaborate on the aggregated content. Within a project, pages must be locked for editing. Publishing an asset is done by checking the page back into what can be thought of as the main branch — a project called “Online.”

Check-in and check-out of any object (template, content, or image) is standard functionality and easily managed using the browser.

Each resource has a history, which means that you can have multiple versions of files. It also enables you to visually compare, see differences between multiple versions, and roll back, if required. Version 7 also introduced a point-in-time page viewing capability.

If you buy the Alkacon OCEE modules, you can set up multiple sites (with delegated administration) that can replicate content back and forth, but this hardly makes the system a candidate for globalization scenarios or multisite management-intensive use cases. Again, it’s not that the system can’t be made to handle these scenarios; it’s more a matter of whether you have the developer time and talent (and budget) to mold the platform to suit your specific needs.

Workflow remains relatively light, but for the scenarios OpenCms targets, we don’t see this as a major shortcoming.

Attributes

Attributes (2) Show Identical Attributes View version 1 View version 2 Print

I	Status	Property Name	Version 1	Version 2
	changed	Date last modified	3/27/06 5:30 PM	7/30/07 11:48 PM
	changed	Size	2697	2637

Properties

Properties (0) Show Identical Properties Print

I	Status	Property Name	Version 1	Version 2
No entries have been found.				

Show everything Show only differences

```

<%@ page session="false" %>
<%@ taglib prefix="cms" uri="http://www.opencms.org/taglib/cms" %>
<cms:include property="template" element="head" />

<h1>Congratulations!</h1>
<h1>Congratulationss!</h1>

<h3>If you're seeing this page via a web browser, it means you've setup OpenCms successfully.</h3>
<h3>If you'reee seeing this page via a web browser, it means you've setup OpenCms successfully.</h3>

(38 equal lines skipped)
  
```

Figure 420. You can visually compare multiple versions in both offline as well as online projects, and roll back, if required. Although each resource can have different versions, there is no way to roll back a complete site.

Experience

Publishing

For content syndication, there is a community-contributed RSS module. This module adds support for RSS and ATOM formats to generate feeds from content items.

Publishing content to mobile and other output devices becomes an exercise in custom JSP development; don't look for built-in multichannel output support (though the reference implementation in version 9 does support responsive design).

By default, the system generates pages with ugly .jsp extensions, but you can fix this through configuration, and alternatively, some OpenCms customers "bake" static HTML pages.

Experience	
Publishing	
Standards Adherence	<input checked="" type="checkbox"/>
Multichannel	<input checked="" type="checkbox"/>
Mobile	<input type="checkbox"/>
Digital Marketing	
Site & Campaign Analytics	<input checked="" type="checkbox"/>
Testing & Optimization	<input type="checkbox"/>
Segmentation & Personalization	<input type="checkbox"/>
Social Media Integration	<input checked="" type="checkbox"/>
Promotional Campaigns	<input type="checkbox"/>
Community & UCG	<input type="checkbox"/>
Workplace	
Collaboration & Networking	<input type="checkbox"/>
Dashboard	<input type="checkbox"/>
Ancillary	
Site Search	<input checked="" type="checkbox"/>
Online Forms	<input type="checkbox"/>
Module Ecosystem	<input checked="" type="checkbox"/>

Digital Marketing

There is no digital marketing, but you could code some bits and pieces using the API.

For example, you could build personalization services on top of the integrated user and permission mechanisms. However, there is no way for business users to create rules for content display.

You can integrate third-party web analytics tags quite readily, and there are modules for social network integration. On the whole, though, if you seek a digital marketing tool, this system won't be a good fit for you.

Ancillary Services

For system search, OpenCms comes integrated with Apache's Lucene Solr search engine. Filters for common file types like Microsoft Word, PDF, and Text are already included, and the default implementation supports excerpt generation and hit highlighting.

There is a module ecosystem with about 30 contributed modules, including some important packages from Alkacon itself. However, many of the modules are older and may not work with the latest version.

Project Intangibles

The company behind OpenCms, Germany-based Alkacon Software GmbH, provides professional services and support. Premium Support, which includes an unlimited number of incident responses (and custom bug fixes), costs €5,000 per year. The only response-time guarantee is "within 24 hours," and even at the Premium level, your calls will be taken only from 9:00 to 6:00 Central European Time, Monday through Friday. Consulting is available through Alkacon.

The core platform — akin to a community edition — is completely free under a liberal LGPL license.

Like the founders of other open source communities, Alkacon has developed what they call Alkacon OpenCms Enterprise Extensions ("OCEEs"). These are commercial extensions for OpenCms, adding features that could be useful for running OpenCms in a medium or large enterprise (or even in a small enterprise: e.g., LDAP connectivity). OCEEs are available as closed-source, proprietary products only. You do not get the source code. Expect to pay about €6,500 here.

Alkacon, for its part, maintains a substantial amount of freely available documentation on its website, but most of it is dated. Nearly a year after launch, the only documentation for version 9 is the online version of the embedded JavaDocs.

The "unofficial" OpenCms forum is available at <http://www.opencms-forum.de>. Here you'll find sample code and numerous discussion forums. In addition, there is a community driven wiki at <http://opencms-wiki.org>, which is quite useful. You'll also find a mailing list at <http://mail.opencms.org/pipermail/opencms-dev/>, but questions can be answered on a somewhat irregular basis.

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

On the plus side, Alkacon hosts a global user conference in Germany each November.

A long list of solution partners around the world can be found online. However, the only criterion for being added to the list is at least one OpenCms reference site, so you'll obviously want to check the credentials of implementers carefully. You'll find the most experienced partners in Germany, where OpenCms deployments are much more prevalent than anywhere else in the world.

Moreover, the fact that Alkacon itself is in the implementation business means that its chances of building a significant developer channel remains hampered. Who wants to compete with the project owners?

Conclusion

OpenCms is a bit of an anomaly; it's a functionally basic system for simple use-cases, but we wouldn't recommend deploying it unless you had experienced Java developers at hand. It features the stability of a platform that has been around much longer than Hippo and Magnolia, but it has never seemed to have obtained critical mass beyond its European base.

OpenCms is a mature, open source CMS best suited to small- and medium-sized departmental web initiatives emphasizing information- rather than marketing-oriented content. Since OpenCms still lacks the fit and finish of a commercial system in many regards, it's best suited for power users who can invest the time to learn its idiosyncrasies. With OpenCms, you are more likely to automate the webmaster role than create a distributed publishing operation. In that regard, OpenCms perhaps most resembles Plone, and likely shares Plone's sweet spot of non-profit and higher-education use cases.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Salesforce: site.com

site.com

Vendor at a Glance

Specsheet	Salesforce: site.com Summary
Geography	Global
What's New	<ul style="list-style-type: none"> • site.com evolves from pre-existing Siteforce product
Strengths	<ul style="list-style-type: none"> • Low-complexity sites can be rolled out quickly • Decent Salesforce integration means relatively transparent CRM data interchange • Editors can use Salesforce Chatter to collaborate on content creation • Customers have access to Salesforce's own CDN • Company promises updates three times a year
Weaknesses	<ul style="list-style-type: none"> • Extraordinary lack of library services (versioning, version control) makes system more of an authoring service than a management platform • Absence of workflow and overly simplistic access model make the platform ill-suited for all but the smallest editorial teams • Employs a proprietary rich text editor, which could prove finicky • Lack of native services for content reuse or multisite management render the platform cumbersome for larger enterprises • Development environment is dated and arcane • Salesforce does not like to provide service level agreements
Potential Fit	Simple Informational, Microsites
Unlikely Fit	All Mid-range and Complex Scenarios
Compare To	WordPress, CrownPeak, Upland Software
Operating Systems	N/A
Repository	N/A
Client	Browser
App Platform	Force.com
Licensing	<p>Software as a Service: From \$19,500/year for a single site with single user access</p> <p>Additional users: \$1,500/year per Publisher, \$240/ year per Contributor</p>
Ownership	Public (NYSE: CRM)

Summary

Salesforce’s increasingly pervasive collection of SaaS-based business tools now includes “site.com,” an extremely lightweight and rapidly evolving entry-level WCXM offering for existing customers of its popular CRM platform.

Salesforce targets site.com for customers wishing to produce marketing and event-driven “outreach” websites rapidly in order to drive business development. The platform is expressly designed with the “field marketer” in mind, helping them author content around web-based campaigns and tying resulting leads back to their Salesforce CRM instance. Nonetheless, remember that other WCXM tools integrate with Salesforce CRM as well, and site.com does not enjoy any special intimacy here.

The real appeal to site.com for early adopters seems to be the fact that licensees haven’t needed to involve IT departments very much (if at all) for deployment. Just remember that you will pay a price for this simplicity. Salesforce has built site.com around time-to-market at the expense of basic WCXM services. The list of missing management capabilities could fill a long rap sheet: no workflow; no version control; no multisite management; no native segmentation; no globalization; no public-facing community applications; and on, and on, and on.

To be fair, these are early days for site.com, and if you’re more adventurous, you could build custom extensions on the force.com platform (although that somewhat defeats the purpose of ease and simplicity). Other partners also can develop modules. Salesforce has poised site.com to develop an ecosystem of its own, via the vendor’s existing “AppExchange.”

In the meantime, if site.com doesn’t meet your specific requirements, then you are heavily reliant upon the vendor’s aggressively planned release cycle of 3 major releases per year to extend its slick — but limited — functionality.

Scenario Fits	
Simpler Site	
Informational	<input checked="" type="checkbox"/>
Microsites & Landing	<input checked="" type="checkbox"/>
Mid-Range	
Basic Digital Marketing	<input checked="" type="checkbox"/>
Mobile Site	<input checked="" type="checkbox"/>
Community Oriented	<input type="checkbox"/>
Complex Site	
Advanced Marketing Portal	<input type="checkbox"/>
Global Enterprise	<input type="checkbox"/>
Multichannel Publishing	<input type="checkbox"/>
Ultra-Large Single	<input type="checkbox"/>
Digital Workplace	
SMB/Departmental	<input type="checkbox"/>
Global/Enterprise	<input type="checkbox"/>

Introduction

Site.com is the latest moniker that Salesforce has bestowed on its SaaS-based WCXM offering, previously billed as both “Siteforce” and “Salesforce.com Sites.” While site.com itself was announced in March 2012, its predecessors (and apparent parents) were first unveiled by Salesforce in late 2010. As of this writing (July 2012), site.com constitutes the spring 2012 version of the renamed existing products, with some new functionality.

The product itself began life as “Sitemasher” from a WCXM startup operation based in Vancouver. Sitemasher demonstrated some early GUI-driven Salesforce integrations and launched its own cloud-based version of the product in mid-2008 on the Azure platform, winning a “Blue Sky” innovation award from Microsoft the same year. Sitemasher was quietly acquired by Salesforce a little over 2 years later.

The original intentions of Sitemasher — to provide a quick and straightforward way for non-technical users to produce interactive websites — are apparent in the platform that grew post-acquisition. Site.com removes all of the typical technical or semi-technical tasks from system users, instead attempting to drive as much as possible from the repository via metadata (similar to how WordPress uses tags to construct pages from metadata applied to individual posts). It is only when users start interacting with data from Salesforce (“Salesforce Custom Objects”) that they are taken from under the bonnet of the platform and require something more than basic web skills.

Salesforce is evolving the platform very rapidly, and expects to produce 3 major updates per year, as it does with its other products. This is a considerable rate of change for any software product, and perhaps all the more so here, given the tremendous ground site.com needs to make up. Early adopters should brace themselves to take advantage of new functionality as it becomes available — and hope that support and documentation can keep pace.

Technology

Architecture

Salesforce is a multi-tenant SaaS solution, with hosting outsourced to collocation companies including Equinix. Most server “pods” reside in North America, but the company has also placed servers in Asia-Pac and (most recently) Europe.

Note that SFDC pre-dates most modern cloud (IaaS) architectures and therefore may not have the kind of elasticity and redundancy that you might come to expect from a purer IaaS platform like Amazon EC2. In the case of site.com, Salesforce makes up for this in large measure by bundling in its own CDN for global content distribution.

Technical Administration and Security

Site.com is designed upon the principle of hiding much of the technical elements of the platform away from users, instead providing simple interfaces to build and present content. This approach has its pros and cons. It does reduce the time required to provision the system initially; customers have praised the product for enabling them to be up and running very quickly after purchase. This also reduces the amount of vendor services usually required to install and set up a typical WCXM system.

The flip side — as with any hosted platform — is that you need to get comfortable with your assets being stored outside of your organization. Site.com does not provide tools to automate local site backups (you are able to export entire sites), so if you’re planning to use site.com as your single repository for web content, give some consideration to local disaster recovery.

Indeed, there is more than a paucity of traditional technical administration functionality available in the system.

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

Performance

With respect to the authoring environment platform, it has historically experienced unpleasant downtime episodes, as well as occasional performance shortcomings. Severe outages appear to be a thing of the past, but the company allows itself comparatively generous maintenance windows and customers still complain about performance. On the plus side, you can track the previous 30 days’ performance and uptime at <https://trust.salesforce.com/trust/status/>. Users have talked favorably about the performance of the administrative interface as well as the rapidity of content deployment. However (as with all cloud systems), it is important to test this within your own organizational infrastructure to see whether it meets your required standards.

Salesforce has given consideration to site delivery performance by utilizing a Content Delivery Network (CDN), meaning that pages are served up by a geo-located version close to where the browser requests the pages. This is a nice touch you won’t find with other SaaS-based WCXM tools, which force you to contract with a commercial CDN supplier like Akamai.

When you publish a page, it gets automatically replicated, which does add a time factor between releasing an item and it being available. Salesforce suggests that this is “fairly simultaneous” and typically takes only a few seconds to take place. Test this carefully — particularly if you need to remove (a.k.a., “flush”) content rapidly as well.

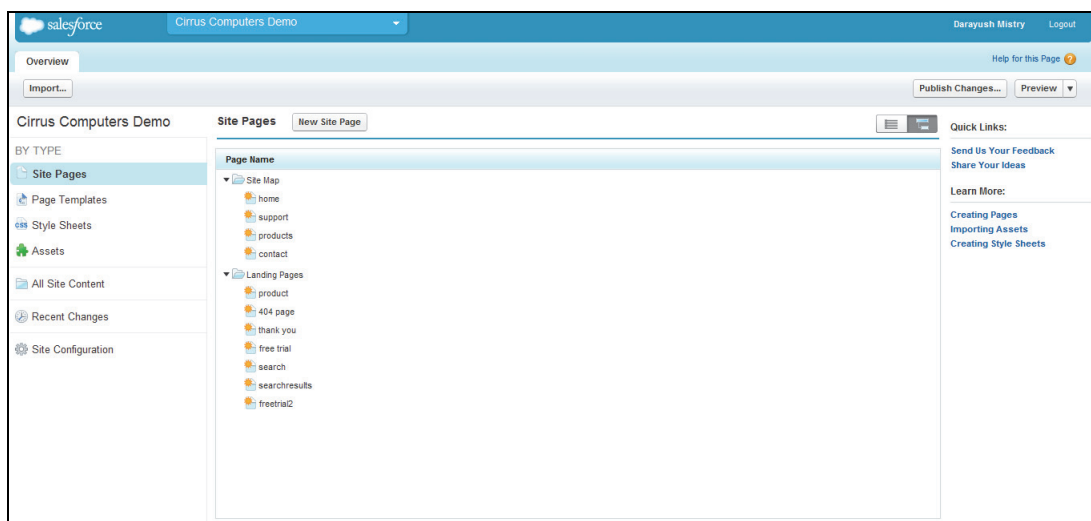


Figure 421. The site.com Studio interface allows users to build cascading style sheets (CSS) inline.

Content Modeling

Site.com has a simplistic approach to content; it is either “Pages” or “Assets.” Pages are associated on a one-to-one basis with page templates, making proper content reuse in this manner impossible. Assets are any additional binary element uploaded to the system (images, documents, or multimedia files). Salesforce warns against using site.com for storing and presenting audio and video assets, suggesting instead that you use an external specialist serving technology (their own developer training material sits on YouTube). This makes site.com undoubtedly limiting to all but the smallest of web properties.

Security & Access Control

Security is similarly simple and limited in functionality. Users are associated with three available roles: Contributors, Developers, or Publishers. The Contributor role has access to edit textual content and place images only; Developers have access to all elements of the system (apart from actual publishing tasks, which are reserved for that eponymous role). Aside from this, there is no way to restrict access to content as you can in most content management systems. Every Contributor can access and edit every piece of content within the site. Publishers can only limit this to a modest degree by locking down areas of the pages via the templating system, allowing only some regions to be editable. In short, you probably don't want to use this system if you have more than a handful of trusted editors. This lack of content governance features has been called out by many users, one commenting that this flaw "make(s) distributing the site content editorial responsibilities [...] dangerous."

Note that these roles factor into site.com's pricing metrics. A Contributor is priced at \$20 per month; a Publisher at \$125; the newly announced developer role is currently un-priced. For a web team of any size, the costs of these "power-user" roles will quickly escalate.

Development

The simple nature of the system necessitates very basic developer tools. Site.com provides users with a menu-driven list of elements to build templates. This will be familiar if you learned how to build websites using tools like Microsoft "Front Page."



Figure 422. Site.com's development environment with very basic developer tooling.

This approach might appeal to those with no HTML skills; however, it does limit you to using only those elements made available by the site.com editor, and relying on the quality of outputted code as best practice and performant. In all likelihood, contemporary web developers will construct their own code outside of this interface and import it to create basic templates. However, while some users have had success taking pre-existing HTML5 code and tweaking it within Studio to rapidly build up templates, others have found it highly problematic. Common complaints are that the parser often rejects legitimate and compliant

code created externally, only to accept the same code when manually edited in the Studio interface.

Similar conditions exist with regard to styling. Studio allows users to build style sheets inline using a basic interface — considered far too basic by many users — for use on an individual site.com site. Again, it is likely that an external tool would be preferred by web designers and as with HTML, style sheets can be imported. However, similar concerns have been expressed with the quality of the site.com parser, and elements have been inaccurately rendered or ignored entirely during the importation process, forcing them to be re-coded manually within Studio. Some users report that the parser can even change the syntax of imported CSS — or only partially parse it — if it encounters an unfamiliar tag.

Both templates and style sheets are wedded to a single site only and cannot be formally copied from one to another. This means that in order to reuse previously developed code, you must clone entire sites, or fall back on copy/paste.

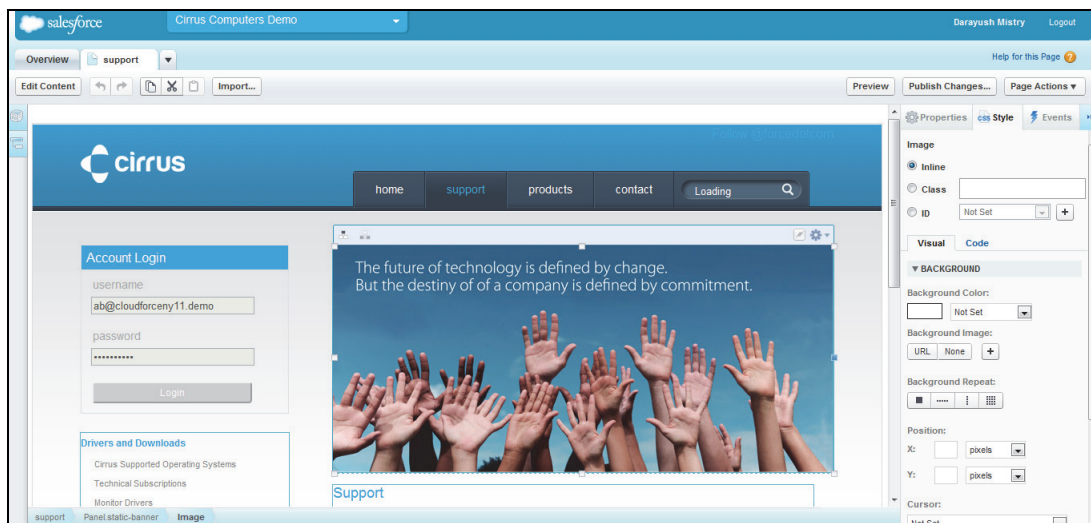


Figure 424. A site.com Studio styling interface.

In order for contributors to be able to do basic content editing, developers must enable “Editable Areas.” These areas are created at a template level, so all pages using that template will have the same editability when created. This means that content types are directly bound to specific layouts (a common limitation at this tier). Additionally, templates can be nested within each other, which should give a degree of flexibility if planned well enough prior to initial development. Note that if you clone a template with nested sub-templates within it, only the parent is cloned. This means that any sub-templates will need to be re-created, since they cannot be copied from one parent to another. Thus, it seems to be of paramount importance to plan how you will template your site in some detail prior to starting work.

As previously noted, there is little in terms of governance features in site.com. The absence of versioning and configuration management more generally means that you can’t really take an iterative approach to building and testing. The development of templates and styling likely will involve a high degree of copy/paste activity between the “Studio” interface and developers’ desktops. Any hope of building even basic source control is going to have to take place entirely outside the product — far from ideal, especially for remote or disparately spread web teams.

Site.com’s integration with Salesforce is its most attractive quality. You can pull content from Salesforce records using “custom objects” and embed tabular data in pages using “data repeaters.” If you store large amounts of company information in Salesforce, this could offer a shortcut for populating certain types of pages. Of course, this would require that the site publisher must have a good working knowledge of the structure of the CRM object data model. The data elements are dynamic, so changes to the structure of Salesforce itself has a knock-on effect to what displays on the site. You’ll therefore want to build some form of “change control” between your Salesforce implementation and site.com to ensure that changes to the former don’t inadvertently break the latter.

Forms built in site.com can be used to populate records back in Salesforce and this is the number one reason cited for choosing site.com. This enables data supplied by visitors to the site to be recorded directly within the Salesforce CRM, which then can trigger other business-focused activity. Customers report that the ability to tie transactions directly to an action performed on a site.com website helps them prove the value of their investment in the platform and subsequent investments in additional sites. Because additional sites are another metric in how Salesforce charges for site.com, it is important to ensure that this dependency between CRM and website works for your organization.

Of course, you can always try to extend the platform. Salesforce has famously evolved into a PaaS provider with its Force.com platform. In theory you can customize parts of the Chatter experience and extend it with custom applications or third-party modules. In practice however, you’ll want to weigh carefully whether you want to wade into the software development business or not. (To be fair, we issue similar cautions about platforms like Jive, SharePoint, and Oracle WebCenter.)

Salesforce has its own proprietary language called “Apex,” which resembles a kind of mash of Java 5 and SQL. It comes with its own Eclipse plugin that is also upgraded three times a year. Developers do not find the platform particularly modern or friendly, and it generally lacks advanced tooling. If you get serious about custom apps or extensions, you’ll want to find an integrator with substantial Apex experience.

Content

Contributor Experience

The site.com interface has been praised by many customers for its usability and simplicity. However, some equally complain that this simplicity comes at the cost of subsequently limited functionality. Depending on the planned complexity of your site or your web team, this could be a boon or a distinct problem. Just remember that the site.com interface cannot be customized by users in any noticeable way beyond very basic role segmentation.

Site.com Studio is a thin client accessed via a web browser, with Firefox or Chrome suggested for the best experience. Internet Explorer is not

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

supported for some of the “drag-and-drop” functionalities within Studio, which might be a limiting factor for some organizations.

Contributing Content

Site.com’s interface for contributing content is functional albeit basic. The company built its own WYSIWYG editor, which typically leads to long-term problems. It lacks many of the features that you would expect to find in a contemporary rich text editor such as the commonly utilized Tiny MCE. Site contributors by default don’t have an access to site pages; it is site publishers that have to create and authorize certain regions — called “Content Blocks” — within pages that are editable. It is only these elements that the “Contributor” user can access.

Some users report that Studio is occasionally obstinate when dealing with page objects created outside the interface — i.e., non-native objects. For example, importing chunks of HTML to allow the embedding of a video can be problematic; they cannot be parsed correctly in the interface, and they cannot be tested until *after* publishing. Therefore, visual tweaks cannot be done, and it would require some code-level intervention. That’s a bummer.

All textual content is edited within a WYSIWYG environment, so contributors can see how their work will look on the finished page with some level of comfort. As always, levels of accuracy with tools of this type vary and should never be used as a guide to “pixel-perfect” positioning. Studio provides some rudimentary text layout and linking tools; however it does not contain a spell-check function.

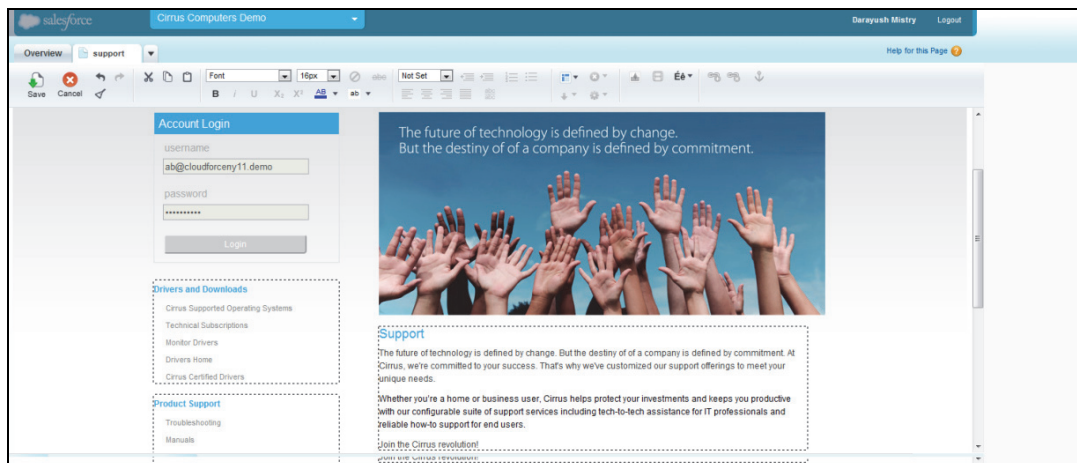


Figure 426. A site.com Studio content editing interface.

Because of the limited capabilities that are assigned to the Contributor role, users that require access to maintain navigation structures or need to create logical folders for content in the system must be assigned the larger (and more expensive) Publisher role.

As previously noted, the lack of basic content governance features — such as check in/out — is concerning. The issue here is that without any locking mechanisms, you run the risk of multiple users editing the same piece of content simultaneously, thus overwriting each other and losing changes. A small, centrally located team could work its way around this, but it is doubtful whether this solution would scale out to larger environments or virtual teams without considerable effort and communications.

Content Lifecycle

Site.com presently has no real concept of a content lifecycle, as it lacks any workflow capabilities and audit functions. Additionally, the platform does not offer rollback or versioning — for content or for templates — meaning as one user put it, “...when they’re gone, they’re gone.”

Content created within one site.com site cannot be formally shared within another site.com site. It is possible to clone an entire site and reverse engineer the site elements to create a similar experience; however there is no inheritance model for this to support a common code base across a range of sites operated by the same web team. The only way in which content can be common is to use the integration points within Salesforce CRM itself, populating the pages using custom data objects. The company says this problem is addressed in a feature they call “Site Networking,” but we saw no evidence of this in any demo, forum, or documentation. It may be a roadmap aspiration. Test carefully.

Similarly, multi-language sites at present require that multiple versions of the same pages must be created and managed separately. Given the overall lack of workflow capabilities, there is no way to link these pages together logically to trigger updates when one language variant changes. If there were an event system that could be used for such capabilities, it is well hidden and does not appear in the current developer documentation.

Experience

Publishing

URLs within site.com are based upon the location within the virtual file structure of the site. This provides the opportunity to apply some logic to how URLs are formed — if the structure of the site itself is well-planned.

Mobile support is limited to CSS-level detection and display, which is not specifically part of the product; it’s up to your designers and developers to control.

Digital Marketing

By focusing primarily on Salesforce automation and customer support (i.e., CRM), Salesforce has traditionally turned to partners for marketing-oriented functionality. Indeed, they have put only scant digital marketing services into site.com.

There is currently little available in the way of segmentation or personalization for site.com.

Additionally, there are no analytics available out-of-the box. Third-party analytics tags (such as Google Analytics) can be added to page templates during development and customers can take advantage of the ability to gather new contacts via the site.com web forms to Salesforce integration.

Experience	
Publishing	
Standards Adherence	
Multichannel	
Mobile	
Digital Marketing	
Site & Campaign Analytics	
Testing & Optimization	
Segmentation & Personalization	
Social Media Integration	
Promotional Campaigns	
Community & UCG	
Workplace	
Collaboration & Networking	
Dashboard	
Ancillary	
Site Search	
Online Forms	
Module Ecosystem	

Site.com social elements rely on embedding third-party components into pages to display content from social media channels such as Twitter. Note that you cannot expose Chatter services to the general public. Commenting or other community aspects are not provided with the platform, which is surprising for a product purporting to allow sites “to be built at the speed of social.”

Ancillary Services

No site search is provided with site.com.

Salesforce has developed a rather large module ecosystem for its other tools, but for site.com the offerings remain sparse.

Vendor Intangibles

Still in its very early days, site.com has a limited community and reach. The platform’s immaturity is reflected in some of the discussions on forums: users report a range of hefty bugs, which are ultimately fixed. Some complaints from users are not so much about the bugs, but more about missing functionalities.

Intangibles	
Vendor Services	
Vendor Professional Services	<input type="radio"/>
Channel Partner Services	<input type="radio"/>
Support & Community	<input type="radio"/>
Strategy & Roadmap	<input type="radio"/>
Viability & Stability	<input type="radio"/>

Salesforce provides basic documentation, which (given the comments on their support forum) often proves to be inadequate when dealing with anything but the most basic features.

Salesforce admits that a rapid amount of development is required to back-fill many of the shortcomings discussed here. However, many of these shortcomings will require considerable overhaul of the underlying data model — the database that underpins site.com itself — in order to resolve them (such as the issues with inheritance and content sharing).

With a commitment to three major releases a year, Salesforce’s plan to have an active marketplace for third-party applications to boost site.com’s functionality seem (at least at this stage) very ambitious. Firstly, there needs to be a critical mass of users on the platform before developers will invest in the market. While there are reportedly some 30,000 websites already running on Salesforce’s infrastructure, the vast majority of these are powered not by site.com, but by force.com, its older and more functionally rich development platform sibling. Secondly, with the level of change in the product — especially where fundamental changes are required to supporting layers of site.com — the lack of functional stability is likely to discourage developers from building on top of what is a rapidly moving target (with an uncertain roadmap).

Conclusion

As of this writing (July 2012), site.com is at a very early stage in its life and this infancy is apparent throughout. It is suitable for building and maintaining very simple sites and its parentage is sure to make it an attractive proposition for some agile web strategies (such as creating perishable microsites).

However, site.com’s inability to provide even some of the most basic tenants of content management makes it really only suitable for authoring very simple sites with an added benefit of Salesforce CRM integration. Even if your requirements match that scenario, the lack of any

real governance or auditing around the management of content or users means that only very small teams will be able to use the platform with any comfort.

Ultimately, site.com is a basic content authoring system rather than a content management platform. Doubtless that will change as Salesforce upgrades the platform, but weigh carefully whether you'll want to go along for that ride.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Telerik: Sitefinity

telerik.com

Project at a Glance

Specsheet	Telerik: Sitefinity 5.1 Summary
Geography	Primarily North America with some UK, Netherlands, and Australia
What's New	<ul style="list-style-type: none"> • New Site Sync feature provides improved content deployment options • New email marketing tools have some segmentation functionality • Visual Studio integration and ASP.NET MVC support
Strengths	<ul style="list-style-type: none"> • Exceptionally low starting price • Cross-browser compatible WYSIWYG editor • Comes with a slew of additional application modules • Very shallow learning curve for .NET developers • Functional administration interface (especially for this price range) that's comparatively simple to learn • Company upgrades functionality on a monthly basis • Nice features for feed aggregation and syndication
Weaknesses	<ul style="list-style-type: none"> • Relatively immature product in terms of feature richness • Sub-standard repository and information lifecycle services • Coupled management and delivery environments may not appeal for some IT departments • Sitefinity has seen potentially serious back-end performance issues, especially with large sites • Weak user management and reporting features • Weak native services for separating development, staging, and production environments reduces enterprise applicability • Templating requires developers • Company does not offer real-time, 24/7 tech support
Potential Fit	Informational, Community-Oriented Site, Microsites & Landing Pages
Unlikely Fit	Advanced Marketing Portal, Global Enterprise, Multichannel Publishing
Compare To	Ektron, Sitecore, EPiServer, Kentico, DotNetNuke
Operating Systems	Windows Server
Repository	MS SQL Server, MySQL, and Oracle
Client	Browser (Silverlight required for analytics module)
App Platform	ASP.NET
Licensing	Commercial license starts at \$499 per domain
Ownership	Privately held

Summary

Sitefinity CMS is a relatively low-cost content management offering from Telerik. Not surprisingly for a company with a long history of developing ASP.NET user interface components, the product explicitly caters to .NET developers. What is surprising for a developer-centric product is the simple user interface that gets end users' attention. Less attractive, though, is the comparative shallowness of Sitefinity's default functionality, especially for larger organizations with more complex processes or regulatory requirements.

Based in Waltham, Massachusetts, and Sofia, Bulgaria, Telerik's model to leverage a Bulgarian engineering team enables them to offer the product for the comparatively low cost of \$499 per domain for their "Small Business" Edition.

Sitefinity's .NET focus, low-cost, and relatively simple functionality make it a logical — albeit less feature-rich — competitor of Ektron. Sitefinity is best suited for simple to mid-range sites, such as Informational Site, Microsite, Basic Digital Marketing, and SMB/Departmental Digital Workplace.

Scenario Fits	
Simpler Site	
Informational	<input checked="" type="radio"/>
Microsites & Landing	<input checked="" type="radio"/>
Mid-Range	
Basic Digital Marketing	<input checked="" type="radio"/>
Mobile Site	<input checked="" type="radio"/>
Community Oriented	<input checked="" type="radio"/>
Complex Site	
Advanced Marketing Portal	<input type="radio"/>
Global Enterprise	<input type="radio"/>
Multichannel Publishing	<input type="radio"/>
Ultra-Large Single	<input type="radio"/>
Digital Workplace	
SMB/Departmental	<input checked="" type="radio"/>
Global/Enterprise	<input type="radio"/>

Introduction

Telerik is one of the younger vendors in this report, having been around only since 2002. After focusing almost exclusively on its .NET designer and editorial controls for the first three years, the company introduced the Sitefinity CMS in 2005.

Of Telerik's 550 employees, 80 are located in North America (Waltham, MA, Houston, TX, and Hudson, OH), while the remaining are at the headquarters in Sofia, Bulgaria with other offices in Germany, India, the UK, and Australia.

Technology

Technical Administration and Security

Sitefinity is based on Microsoft ASP.NET and uses ASP.NET concepts like controls, membership, and profiles. Hosting it requires Windows server 2003, 2008, XP, Vista or Windows 7. It also requires the IIS web server (minimum required is IIS 6+ but it's better to have IIS 7+), and .NET Framework 4.0. Sitefinity utilizes other Microsoft technologies such as Windows Communication Framework (WCF) for Web Services, along with Silverlight and Windows Workflow Foundation.

Even though you can run it in a shared hosting environment, some of the requirements above could make that a little tricky. If you plan to host in a shared environment, make sure your host has the proper pieces in place.

Sitefinity is deployed most commonly as a bundled production and delivery system. Most Telerik customers buy only the standard license, which only allows the administration and public environments to run off one server. Larger enterprises — or those with high traffic requirements — will need to investigate the Professional or Enterprise version that (in theory) supports load balancing. This load balancing is dependent on Microsoft's "Network Load Balancing Services" (NLBS, an optional element within the Windows Server product range), which must be enabled so that an active cluster can be monitored for availability.

Added in version 4.4, "Site Sync" is Telerik's attempt to address its weaknesses in content deployment that has long been apparent in the platform. According to the vendor, Site Sync allows you to create multiple deployment targets — such as staging and live — and content to be deployed is based upon some administrative logic. However, you should carefully test these claims, as content deployment remains one of the major weaknesses of the product. For example: you will be required to do further content synchronization between staging and live via variations of time-based triggers, or using manual push jobs. In addition, this "all-or-nothing" process cannot be managed on an item-by-item basis. Even with detailed configuration, it is limited to certain content types (e.g., sync all press releases).

Overall, Sitefinity in this area is far away from the granularity of other deployment tools. Compared to Joomla!, DotNetNuke, and to some degree Kentico offerings, Sitefinity has a poor grasp on the concept of staging. Unless Site Sync has been configured to create a pseudo-staging environment, once an item is live, authors must make changes to the content directly in the live environment. This kind of publishing model works for small setups because it provides real-time previewing and easy publishing. However, it may not be suitable for larger implementations that require more formalized publishing processes, security, and a separation between editorial and live environments.

From a licensing perspective, keep in mind that Site Sync is not available in the Small Business or Standard Editions. Your Sitefinity investment will have to start at a considerably higher licensing bracket if you want to use Site Sync.

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

Sitefinity targets .NET developers. More so than Ektron or Sitecore, Sitefinity applies generic ASP.NET controls and methods, making the product perhaps less out-of-the-box, but more comprehensible to developers. In fact, customers report that developers with a small amount of .NET experience find Sitefinity's architecture easy to understand, and they typically get up to speed in a matter of days, not weeks. Few other products in this report can boast that claim. Bear in mind however, if you do not have experienced .NET developers on your implementation team, Sitefinity probably is not the right choice for you.

With respect to the publishing model, Sitefinity is a coupled system and there's only one environment for editorial-stage content, and live content. When you publish content, its status changes to "Published," but everything else remains unchanged. It is also a "frying" system, and pages are generated on the fly when they are accessed for the first time.

Telerik applies the ASP.NET "provider" concept as an abstract repository access model; thus this is a "pull" system, where you develop a custom front-end. The advantage of a "provider" pattern is that it is comparatively easy to replace your back-end repositories without changing your implementation.

Sitefinity has roles, but not groups. Actually, the product has groups, but calls them roles. This simple approach might work for you if you don't have a wide variety of professionals working in the application. By default, you can authenticate against the product's native directory or Active Directory, and you can also apply an ASP.NET custom membership provider, which may be handy in certain cases.

The user profile information is fixed and it is not easy to customize it or add more fields to it. This could be limiting for scenarios that require information to be captured to obtain user profile information.

Sitefinity offers very little in the way of reports. You can see and sort some status indicators, but that's it.

Development

Since Sitefinity is based on ASP.NET, developers can customize or extend it using familiar .NET concepts with Sitefinity APIs (both C# and Visual Basic versions are available). Sitefinity provides a RESTful Web Service API based on Windows Communication Foundation (WCF), which enables you to perform different operations on assets. Additionally, Sitefinity provides other APIs to access Sitefinity functionality.

Templates, themes, and controls are the ingredients that the system uses to fry the pages at runtime. Templates and themes dictate the look and feel of pages; templates define a page's layout and the themes function as style sheets. Sitefinity templates are essentially ASP.NET master pages. The advantage is that if you understand ASP.NET, you can create Sitefinity templates relatively easily without learning new concepts. Alternatively, to create simpler templates, you can use a visual layout editor, although Telerik believes that most of its larger customers will reject this in favor of having dedicated, in-house design skills that use their own tools of the trade, rather than those provided by Telerik.

Those familiar with .NET will be familiar with the notion of controls. Page content or functional logic are contained within .NET controls.

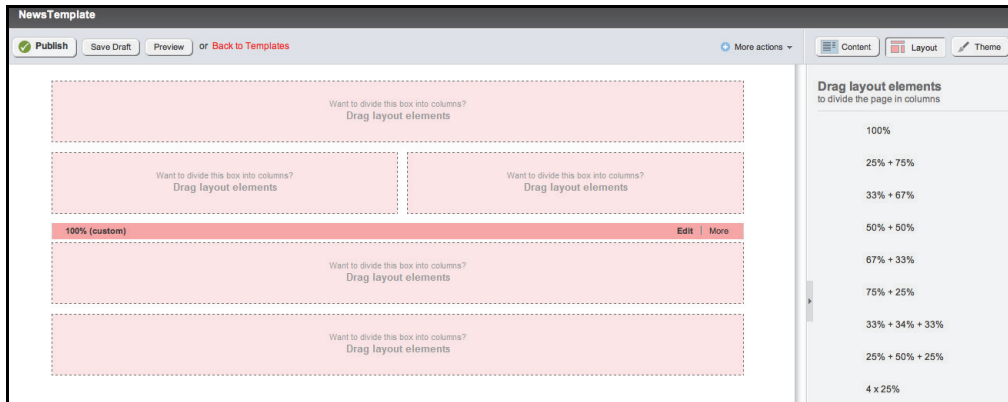


Figure 428. You can use this visual layout editor to create simple templates. For more complex templates, you also can use ASP.NET master pages in Visual Studio.

When it comes to controls (which are historically Telerik’s sweet spot), Sitefinity offers several options. The product ships with some simple controls such as the “Generic Content” control, which simply displays text and images. Sitefinity also comes with Telerik’s suite of user interface controls. If these do not sufficiently meet your needs, you can also build your own controls; however, you will need an experienced developer to do this.

Developers seem to love the freedom and flexibility that Sitefinity provides. One developer described the CMS as “one big API.” This freedom can be both good and bad; too much customization has caused problems when moving to new versions of the product, and customers have warned that it is easy to design the system in a way that provides end users with too much control.

Developers are likely to utilize the Visual Studio plugin, “Sitefinity Thunder,” which allows the management of code and code releases directly from within the IDE. Initially, it purely dealt with Sitefinity “Themes” (essentially, page templates), but more recent iterations have extended the reach into the realm of “custom modules,” which enables database tables and their data validation rules to be created and maintained via the plugin.

Code references — or “Code Templates” as Telerik refers to them — can be pulled into the interface providing some “copy-and-paste” shortcuts. Auto-complete is also available, querying Sitefinity for custom properties inline while creating code. It’s not yet a mature offering, but it seems to be undergoing a pretty rapid development cycle. Thus, it is important to test it out with your .NET developers so see whether this approach will fulfill their needs.

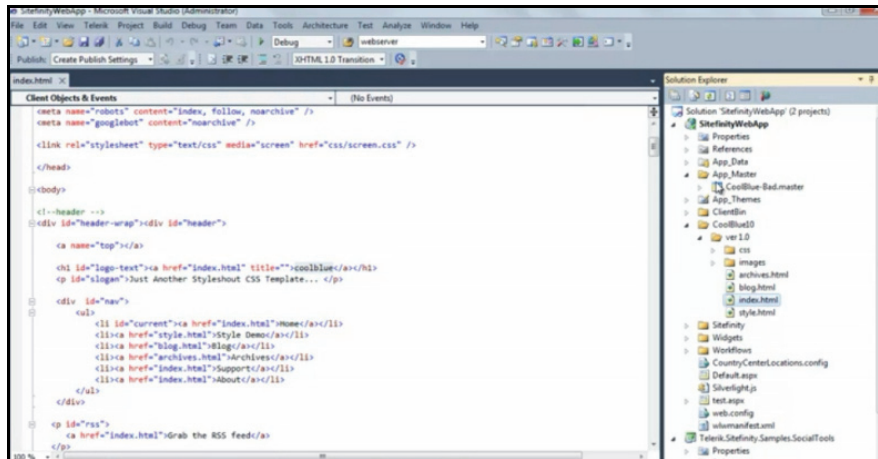


Figure 429. Developers use Visual Studio to create new templates and extend Sitefinity via the “Thunder” plugin.

The debugging features are quite primitive. If you are trying to figure out why a template is not rendering properly or why a particular Web Service is throwing an exception, you are on your own. Developers use external tools and plugins like Firebug (a Firefox plugin). The error messages also leave much to be desired.

Performance

Some customers we spoke with indicated experiencing periodic performance issues within the back-end dashboard interface. One customer with a site with over 9,000 pages claims that since the site has grown, the site has slowed to the point where members of his team avoid using the dashboard. If you have a large site or plans to grow your site, consider this carefully.

As with all “frying” systems, you need to be wary of performance in high-traffic situations on the delivery side. Sitefinity offers a page-caching capability, which can help, but a few developers have indicated that this can be tricky to navigate. You'll need to stress test this area carefully.

Sitefinity version 5.1 added support for the ASP.NET MVC — a Model View Controller-based web framework that (in principle, at least) should offer some performance advantages on the delivery side of the product. Telerik has opted to continue supporting Web Forms alongside MVC. The two can even be combined in templates using a “mixed mode” setting. Given that there are situations when one or the other is more appropriate (for example where session state maintenance is essential, and Web Forms still remains the best option), this is probably a pragmatic decision.

Historically, when setting caching settings for Sitefinity sites, customers must come up with one setting that would work globally for all pages. In response to customer requests, Telerik added a much more useful capability to apply caching durations at the page level (in addition to the site level). However, unlike higher-end (and pricier) systems, there is no component (or widget-level) caching, so dynamic or personalized pages may still take a performance hit. For distributed environments, caching is dependent upon utilizing the load balancing features within the Professional or Enterprise version of the product. Load balancing is not included in Small Business or Standard versions of Sitefinity.

Content

Contributor Experience

The default interface for content contributors (and all other back-end users) is “Dashboard.” Dashboard only displays the features that you have permission to use. If you are a content contributor, you will have access to the functionality that’s related to creating pages and content. If you are a power user or an administrator, you use the same interface, but you have access to more functionality.

The power users with whom we spoke almost unanimously like the drag-and-drop functionality that easily enables them to arrange and reposition page elements. However, casual contributors who do not use the system on a regular basis may have difficulty grasping the concepts of templates, themes, controls, and modules. The product does ship with contextual help to aid with these conventions.

The Sitefinity user interface supports Internet Explorer, Google Chrome, and Firefox, but it requires a Silverlight plugin if you plan to use the analytics reporting feature. The Sitefinity UI is not tested on the Mac OS, and the company has no immediate plans to support it.

Sitefinity currently ships in nine languages: English, Portuguese, Dutch, German, French, Turkish, Arabic, Spanish, and Italian. All of the interface labels, help text, and error messages are stored in resource files that can be localized into other languages; however, you need to do that work yourself. For the supported languages, these localization must be downloaded and installed on a “need-to-use” basis, and Telerik suggests keeping them in sync with each release of the product. You might run into problems with separate modules lagging behind in terms of the latest localization updates, which means they may not be compatible with the latest version of the overall product. Choose your modules carefully.

Contributing Content

Similar to its close competitor Ektron, Sitefinity comes with its own WYSIWYG editor. Telerik's RadEditor for ASP.NET uses AJAX to create a very Word-like user experience. End users particularly like the in-line, Word-like spell checking. Its cross-browser compatibility is often lacking in other WYSIWYG editors. It supports all of the major browsers, including Safari and Opera (though the broader interface is only certified on Windows). It also offers an impressive real-time XHTML validity check through a W3C service.

Sitefinity offers in-context editing and you can make changes while browsing the website. However, when you actually edit the content, it opens the back-end form in the same window. Thus, even if you need to change a simple headline, you must open a complete form. If you use a Mac, make sure you test this feature well, since it’s not officially supported.

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	



Figure 430. Sitefinity supports in-context editing. Note the black edit boxes that display with editable regions.

Authors can build custom pages by dragging and dropping controls, which is slick, though you have to be careful about creating reuse situations that become difficult to track later (more about that below).

You can use hierarchical categories or informal tags to organize your content. These categories and tags can be used to query and display content.

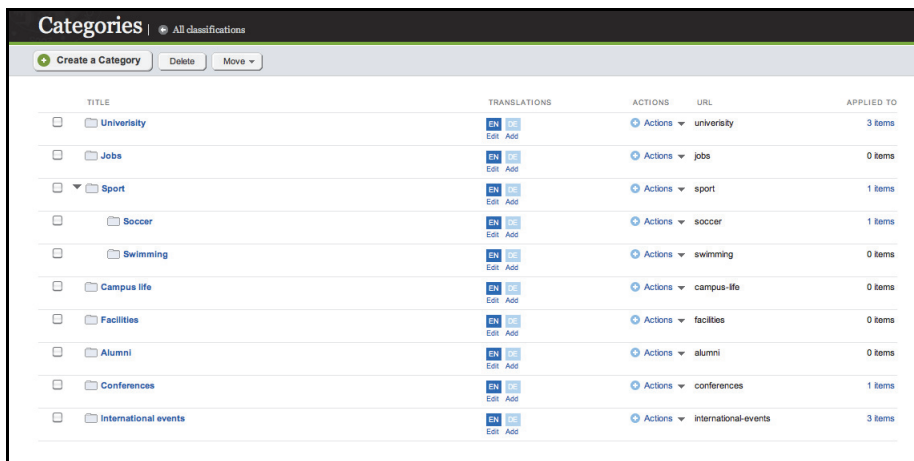


Figure 431. Categories can be hierarchical or you can use flat tags.

Overall, the UI for content contributors is functional but appears dated. Content creation happens via a simple forms-based interface, without some of the advanced features that many other tools have. For example, there's no automatic type-ahead capability for tagging content, and there's no way to apply categories to a collection of content items, unless they are defined as the same content type (you can do it for documents and images).

By default, Sitefinity is a page-based system. You create a page and then assign content blocks to it. This is fine for scenarios with minimal content reuse, but you start getting into problems once you start reusing content across multiple pages or multiple sites. For example, it is not possible to assign the same content block to multiple pages, although it is possible to share content types via some additional configuration (for example, you could allow all "press release" content types to be shared). When reused content is edited, a warning is given that the content appears in multiple places, but there is no signal where the other instances are and there's no way to visualize how content is shared or reused across a site or multiple sites. This

may make editing difficult on larger sites with substantial content re-purposing. Information governance — difficult even at the best of times — will be a challenge for all, even for the most organized users.

Out of the box, Sitefinity comes with a few content types, including those for managing binary assets such as images, documents, and videos. These offer basic metadata support, and can point to assets stored in the Amazon cloud. You can add additional fields to support your own metadata.

If the out-of-the-box content types are insufficient, or if you need to add additional fields in those content types, you will need to create a new module. This would require developers to create XML files to hold content type information and modules for them to display.

Sitefinity enables users to roll back to any previous version of content, and you can do side-by-side, visual comparisons.

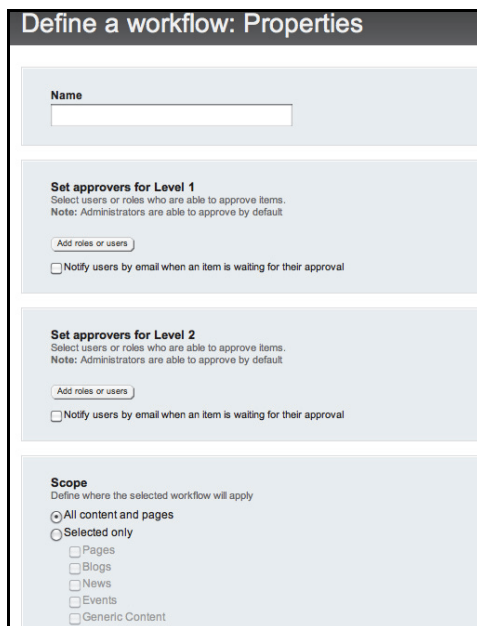


Figure 432. Workflow configuration screen.

Content Lifecycle

In Sitefinity, you can substantially customize workflows (like adding approvers or deciding what content types apply), and create different variants. Usually these suffice for simple use cases.

You can set email notifications and use options that enable admins to bypass the workflow. Alternatively, if the available workflows are not suitable for your use case, you can create completely new workflows based on the Windows Workflow Foundation. However, keep in mind that you will need developers to create those workflows using Visual Studio and ASP.NET; this will not be trivial.

Like most content management systems, content items in Sitefinity have a notion of states (such as Draft and Send for Approval). However, according to customers, once an item’s state is promoted, it cannot be rolled back easily.

The system lacks some standard lifecycle services. There’s no ability to archive content specifically (you can “unpublish” it or delete it). Similarly, there’s no inbox functionality that shows pending items. The workaround is to go to the content listing and filter content based on the status of different workflows.

One nice feature is the ability to sync multiple translations of a page. If you make a change to an English version (such as adding a content block), that change is automatically reflected in the layout of other language versions.

Sitefinity may not be ideal for enterprises that need distinct staging and production environments, because there is no such concept in the product. The product does have a “sandbox” area for development, but this area is only for testing out new modules and code changes. All “draft” content can be previewed in the context of the production environment,

but content creators and reviewers do not have access to a full staging environment where they can view multiple content changes.

The product is not natively designed for multisite management. Part of the challenge here is the “provider model” for content delivery (where the delivery application is) is somewhat unaware of the repository. Many customers with multiple websites seem to set up multiple instances of the tool. Sitefinity will allow you to domain-map different languages to different websites and toggle between them in the editorial environment, but there are no controls to do things like trigger a localization workflow when a source language has changed.

Sitefinity is not a good platform if you need content retention. Some modules are date-aware; others aren’t.

Experience

Publishing

By default, Sitefinity now generates intelligible URLs, and you can configure the package to generate friendlier URLs that apply semantic and extension-less page names.

Digital Marketing

Unlike other WCXM vendors that embed homegrown analytics into their products, Sitefinity has integration hooks into Google Analytics. It is quite nifty to be able to see Google Analytics reports directly in Sitefinity’s interface.

Version 5.1 of Sitefinity introduced a home-grown email marketing platform “Email Campaigns,” offered as an extra, with pricing pegged against subscriber numbers. “Email Campaigns” enable creation of HTML-based emails in the CMS UI, as well as some tracking and A/B testing facilities. Tracking comes in the form of Google Analytics tracking codes. The A/B testing facility allows some basic segmentation based around samples taken as percentages from “winning” results. The winning results can be defined as “opens” or “clicks.” It’s not at the same level of richness or depth that specialist tools offer, but may be good enough for some small businesses.

Ancillary Services

The company continues to add an increasing number of pre-built applications called “modules.” To date, the product comes with several modules that provide administrative controls and a public interface. These include pre-built blogging, wiki, polling, and forum discussion modules that site administrators can drag and drop onto pages.

Another example of a module is “Events” for simple calendaring. These modules are generic, in that they do not cater to any particular industry vertical. As the customer base grows, we

Experience	
Publishing	
Standards Adherence	
Multichannel	
Mobile	
Digital Marketing	
Site & Campaign Analytics	
Testing & Optimization	
Segmentation & Personalization	
Social Media Integration	
Promotional Campaigns	
Community & UCG	
Workplace	
Collaboration & Networking	
Dashboard	
Ancillary	
Site Search	
Online Forms	
Module Ecosystem	

anticipate pressure from certain industries for Sitefinity to build some industry-focused features. We interviewed one media customer whose requests were “email to a friend,” “print this page,” or “save to PDF.” This type of functionality is commonplace in products that cater to media publishers, but it’s unavailable in Sitefinity.

In 2011, Telerik acquired Mallsoft, a San Diego-based ecommerce vendor that previously had been providing an add-on to the Sitefinity platform. This now re-written addition is available as an extra-cost module (it’s included in some of the higher-end Sitefinity packages). It shares a common repository with the core platform by adding a number of extra tables to the repository database. Currently, Telerik claims that there are in excess of 50 customers actively using this module.

Sitefinity offers some nice feed aggregation and syndication capabilities. You can import content into Sitefinity from external RSS feeds or expose content as different feeds. The import and export processes do selective filtering based on field names or sources to aggregate or syndicate subsets of relevant content.

For site search, Sitefinity integrates a modified version of the popular open source search tool Lucene. Business users appreciate the ability to drag and drop search boxes to pages on the site, as well as the ease in which they can determine which areas of the site will and will not be indexed by the search engine.

Vendor Intangibles

This is not one of those products with an aging, stale codebase and feature set. Telerik has developed a reputation as a rapidly developing vendor, and the product has garnered a similar reputation. Dot releases — some with significant functionality — come almost monthly. Many customers welcome this; some find it a hassle.

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

Viability & Stability

Aside from some strange accents in some of the older training videos (more recent videos are voiced by native English speakers), customers have not had a problem with the fact that most of Sitefinity’s workforce is located in Bulgaria. In fact, many North American customers like to submit work requests to Telerik in the evening, and often find the work completed by the time they return in the morning.

Telerik guarantees a response to all weekday customer requests within 24 or 48 hours, depending on the license type. This may be too sluggish an SLA for some.

Unlike many vendors, Telerik publishes all of the Sitefinity documentation on its public site. Note that nearly all of the documentation is oriented toward developers and not highly regarded, especially in respect to the underlying API. There are also publicly available user forums, a knowledgebase, and a guided video demo. Customers report that the user forums are quite active. Telerik has recently begun hosting user conferences — the first of which was held in London in 2012 — and promises that this will be repeated from now on, allowing the community to meet personally.

As already indicated, do not undertake a Sitefinity implementation project without the help of experienced .NET developers. For those enterprises without in-house .NET experience,

Telerik has several professional services partners, and has recently set up a marketplace for them to promote their own modules. Telerik recently introduced a training requirement for these partners; Gold partners need to have one developer fully certified to maintain their status; Platinum partners have to have double that as a minimum.

In 2012, Telerik acquired US-based systems integrator NimblePros to extend its professional services reach in North America, primarily for their Enterprise customers. Given the very recent nature of this venture, expect ample ramp-up time for NimblePros as they gain the expertise needed to support Sitefinity implementations. From a staffing perspective, the 10+ team at NimblePros will likely have to grow, acquiring and training new talent in order to provide sufficient amounts of geographical reach and quality support for Sitefinity's Enterprise-level customers in North America.

There are several resources available for potential buyers to test-drive the product. There is a publicly accessible online version of the tool, where you can actually publish content to a test site. It's worth noting that there is a scaled-down version of the tool called "Sitefinity Community Edition," which is available as a free download. It does not have all of the features of the Standard Edition, but it's a good way to try it before buying a Standard license.

Conclusion

Telerik has experienced steady growth in recent years, and Sitefinity is beginning to make a name for itself. For its very low price, Sitefinity offers impressive functionality, although you may still find that its reuse, repository services, and performance attributes are lacking. Nevertheless, Telerik continues to add more modules and features at a highly aggressive pace. In turn, the company also has added more customers. For a product with a history of releasing features that weren't ready for prime time, the impact of rapid customer growth remains to be seen.

The Sitefinity learning curve seems shallow for authors and for .NET developers. Customers like the fact that they can get a site up and running quickly, and that the product provides enough flexibility to extend and meet specific needs. With any product in this price range, carefully examine its scalability in the context of what you are trying to accomplish. If you need things like multiple environments for staging and production, consider other options. For Microsoft-oriented customers with simple scenarios and (especially) lower budgets, Sitefinity could be an appropriate fit.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

TERMINALFOUR: Site Manager

www.terminalfour.com

Vendor at a Glance

Specsheet	TERMINALFOUR: Site Manager 7.1 Summary
Geography	Mostly Europe and North America
What's New	<ul style="list-style-type: none"> • Automated migration and import tool for various CMS packages • Reporting integration with Google Analytics
Strengths	<ul style="list-style-type: none"> • New content types and templates are relatively easy to create • Supports unusually sophisticated workflows • Live previewing of PHP, ASP, and JSP pages • Product produces short, friendly URLs (that can be manually edited) • Unusually strong support for accessibility standards • Very active and open customer community with emphasis on higher-education institutions in UK, Ireland, North America • Company appears to be responsive to feature requests • High-value offering for smaller websites with modest requirements • Interfaces are localized into Welsh, Irish, French, German, & Spanish
Weaknesses	<ul style="list-style-type: none"> • Company knowledge of markets outside of education, membership organizations, government, or financial services is limited • Page-based orientation is not ideal for content reuse scenarios • Surprisingly few native features for multisite management • Although possible to extend, it's not a development platform compared to other offerings and is not ideal for complex scenarios • Peculiar, content-oriented licensing model punishes larger websites • Company appears heavily focused on services and runs a bit like a consulting operation • With a dearth of knowledgeable consultancies in the tool, customers depend on the vendor itself for professional services
Potential Fit	Microsites and Landing Pages, Informational Sites, structurally simple Ultra-Large Sites
Unlikely Fit	Community-Oriented Site, Global/Enterprise Digital Workplace, Advanced Marketing Portal, Global Enterprise
Compare To	Hannon Hill, Percussion, OmniUpdate
Operating Systems	AIX, Windows (including Vista), Solaris, Linux
Repository	Databases: Microsoft SQL, MySQL, Oracle, PostgreSQL
Client	IE, Firefox, Safari, Opera, on Windows/Vista, Mac, Linux, Solaris
App Platform	Java: WebSphere, Oracle WebLogic and OAS, Tomcat, JBoss
Licensing	Based on number of content items; €10-50,000 plus 22% for support
Ownership	Privately held mostly by company principals with some equity financing

Summary

TERMINALFOUR is an Irish vendor and its Java-based Site Manager product is noted for its accessibility awareness and comparatively clean, cross-browser-compatible interface. The company is known for its domain expertise in areas of higher education and government, and has scored sizable wins in those sectors in Ireland and the UK — as well as a smattering of universities in North America. The company boasts a share of “30 percent of the UK’s top 50 universities.” TERMINALFOUR also has conducted a significant amount of business in the financial services and membership areas, such as labor unions.

Until a few years ago, TERMINALFOUR has had almost no penetration of the US market (less than three percent of 2007 revenue), but the company has opened an office in Boston, and says North American billings now account for nearly 40 percent of overall income.

Consider TERMINALFOUR’s Site Manager for scenarios that involve batched bake-and-push content to a web delivery tier outside the firewall. You should also consider Site Manager for microsites that require delegated administration, where its clean, logical, administrative interface and comparative ease of deployment (with ample features for non-technical users) are strong pluses.

In general, this is a low-cost system. However, be forewarned that because of TERMINALFOUR’s unusual licensing scheme (which charges based on content items in the system), customers with large sites built of highly granular content may find the licensing costs to be onerously high.

More generally, you’ll want to understand that TERMINALFOUR takes a very consulting-oriented approach, which is good if you want substantial assistance from your vendor. However, it’s less of a fit if you are looking for packaged software or you already have a close relationship with an integrator that (likely) has no proficiency with Site Manager.

Scenario Fits	
Simpler Site	
Informational	<input checked="" type="checkbox"/>
Microsites & Landing	<input checked="" type="checkbox"/>
Mid-Range	
Basic Digital Marketing	<input type="checkbox"/>
Mobile Site	<input checked="" type="checkbox"/>
Community Oriented	<input checked="" type="checkbox"/>
Complex Site	
Advanced Marketing Portal	<input type="checkbox"/>
Global Enterprise	<input type="checkbox"/>
Multichannel Publishing	<input checked="" type="checkbox"/>
Ultra-Large Single	<input checked="" type="checkbox"/>
Digital Workplace	
SMB/Departmental	<input type="checkbox"/>
Global/Enterprise	<input type="checkbox"/>

Introduction

TERMINALFOUR was founded in Ireland by Piero Tintori in 1996 as a spin-off from his consulting company. By 1999, the company came to focus on Web CMS software, though it still performs a wide array of customizations (and counts sales engineers and consultants as part of its R&D arm). With Site Manager, TERMINALFOUR has found a niche in higher education, government and financial services.

Version 7.0 was released at the end of 2009, with a significant overhaul of the user interface. Since then, the company has consistently pushed out minor releases on an almost bi-monthly

schedule. The 7.1 release came out in June 2011, focusing primarily on new migration capabilities for licensees transitioning from another vendor. (TERMINALFOUR tries to target customers from OpenText, and HP — WCXM tools that have acquired by other vendors.)

Technology

Technical Administration and Security

Site Manager is based on what TERMINALFOUR calls its “T4 Business Platform,” which is written as a J2EE servlet application that will run in almost any J2EE application server. It’s based on an extensible collection of “handlers” that implement specific business functionality (e.g., workflow). Licensees with sufficient Java expertise can extend these as they see fit or develop new ones. The product ships with an assortment of prebuilt handlers, many of them stemming from custom development that’s done on behalf of (or even by) customers.

The latter deserves further comment. Although Site Manager is assuredly not open source, it’s an unusual product in that all of its Java classes are documented and can be programmed against, such that an experienced Java programmer can extend the product in nearly any direction. TERMINALFOUR also claims it provides a Web Services API for Java interfaces. Many customers (and consultants) have done just that, developing new features that often got donated back to the company and ended up in the product’s core codebase — much akin to how community code is folded back into the trunk of an open source project. However, this meant that an increasing amount of product code was developed through an ad hoc process that didn’t conform to the more formalized kind of development process that you would see at a major vendor like Oracle. However, with the funding the company received in 2009, TERMINALFOUR was able to ramp up its own developer resources and now boasts 60+ employees. Version 7 (and a subsequent minor release) appears to be the result of a much more managed process than before.

Underneath the T4 Business Platform lives the “SQL Factory,” a JDBC-based layer that provides platform independence for database transactions. A nice touch for systems integrators is that the product stores system artifacts and content in a database under schemas that are easy to inspect and extend (Figure 433). The content can be queried easily from other enterprise apps; nothing is “hidden” or converted to a proprietary blob or some other arcane format. Everything is out in the open.

Site Manager is partly JCR compliant under JSR 170. However, you can’t swap in a non-T4 repository (even if it’s JCR compliant) for use as a Site Manager repository, but you can access the Site Manager repository from a different application using standard JSR 170 API calls.

Technology	
Technical Administration & Security	
Threat Prevalence	
Authentication & Authorization	
System Reporting	
Multisite Management	
Cloud Services	
Development	
Configuration & Customization	
Integration & Extension	
Content Modeling	
Templating	
Performance	
Back-end Performance	
Site Caching & Delivery	

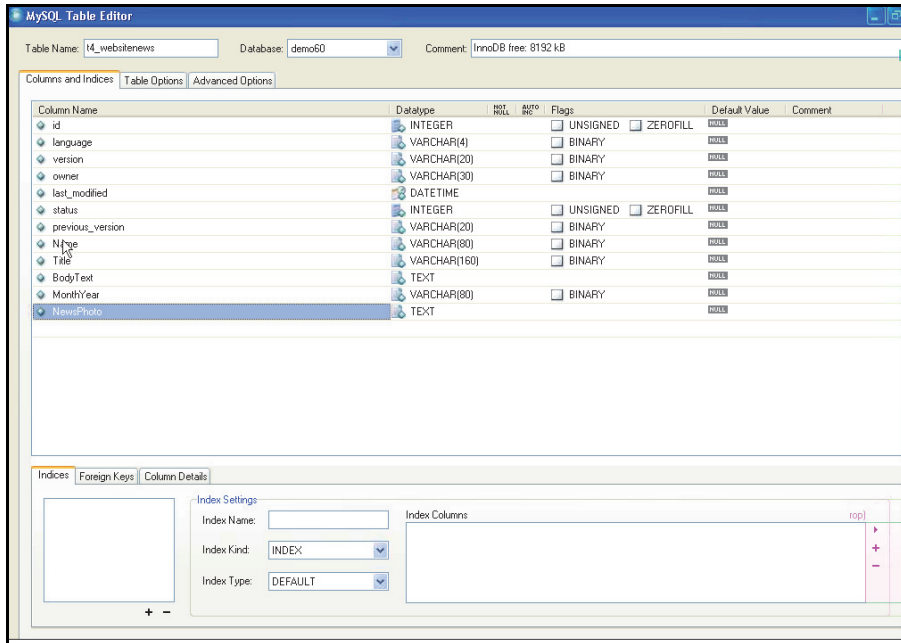


Figure 433. Site Manager stores metadata and content in a database, using schemas that are easy to inspect (as here, using the MySQL Table Editor).

With respect to publishing, this is fundamentally a “baking” system that pre-generates static HTML out of a database, but TERMINALFOUR offers the option of both dynamic and static delivery. A small number of customers opt for a Servlet or JSP on the web server, which is used to display content synchronized from Site Manager’s cache dynamically. Most, however, opt for “baking” their web content and use Transfer Manager to generate the pages, then transfer them (via FTP, FTPS, or SCP) to the web server. In this case, no TERMINALFOUR components are required in the delivery tier.

Like all baking systems, you can “parbake” scripts and special tags (JSP tags, PHP, etc.) into pages to pass through to the runtime delivery environment, so that the runtime web server does the final dynamic assembly of pages. As always, this makes preview tricky in the CMS itself, and you will need to push those dynamic pages to a staging environment for fully virtualized preview. Pages built on ASP, JSP, and PHP can be previewed live in the system, but it’s up to you to arrange for synchronization of content across stages. According to TERMINALFOUR, previewing ASP or PHP pages is not a problem, but test carefully if this is a requirement for you.

Site Manager provides a Scheduler facility that lets you set up automatic periodic deployments (or other regularly recurring operations) from the browser UI.

The system has its own built-in authentication tool, but can also authenticate users in directories using LDAP or Active Directory via JNDI. The system has features for filtering LDAP directories, which is particularly useful for customers in education, where common to have over 10,000 users. T4 also supports Shibboleth and CAS integration, the open source SSO solutions favored by Higher Ed.

Four different roles are provided out of the box: Contributor, Moderator, Power User, and Administrator. The Administrator has godlike rights over the entire product. A Power User has administrative rights over a “channel” (typically a microsite). This is intended to enable delegated administration of the kind you typically see in higher education, where multiple departments have their own microsites and each one wants its own administrative capability (but you don’t want the administrator of one microsite to have admin rights in a different microsite). A Contributor can author content, and a Moderator is a contributor who can make changes to page composition (and typically acts in an Approver role in workflows). The various rights pertaining to these roles can be modified, and custom roles can be created, using the browser UI and (depending on your exact needs) custom Java code. Few licensees seem to customize the default roles, however, since anything more is overkill for most scenarios that Site Manager targets.

From a reporting perspective, you can view the audit trail for users, actions, templates (content types), and other targets over a specified range of dates. There are also reports for SEO and accessibility. Note that these are run as batch jobs on the server; thus, how fresh their data will be depends on the load you are willing to put on the system. A helpful feature is the data export on all pages, sections, and templates to an Excel spreadsheet, because you can run your own reporting macros using them. TERMINALFOUR claims that this has been further improved in 7.1; it can integrate with tools like Google Analytics to display reports in the CMS dashboard.

Note that TERMINALFOUR doesn’t natively deliver direct support for multiple development environments like testing, staging, and production. You would typically publish the content to a testing or staging environment and from there, push it to production.

Development

Website development is mostly up to you. Although the published files can be of any type and therefore can be coded in any language, the web server is unaware of the repository. External data sources can be aggregated into the site at publish time using what TERMINALFOUR calls “Data Objects.” These are almost identical to navigational objects, but they are different because they are accessed through their own section in the interface and they pull data out of other sources and integrate them in web pages (typically through JDBC, but it’s also possible to use Web Services).

The templating system uses “styles,” “templates,” and “navigational objects.” Content components are defined by the templates and consist of elements (fields) and the formatting of different types. The latter can be done using a combination of CSS, XSLT, (X)HTML, and/or XML containing a special <t4> tag grammar that allows for complex substitution of content. Note, therefore, that “templates” not only define the structure of the content, but their rendering as well. Nevertheless, you can also specify multiple renditioning options for a given piece of content. The same content can be output as text, (X)HTML, XML, RSS, and multiple other MIME types.

Pages can be composed of multiple content items, including lists that are dynamically populated. The system offers good flexibility in repurposing of content across pages, but you shouldn’t mistake this for a Component Content Management system that enables construction of complex engineering or legal documents from granular XML fragments.

Unlike some other products that are unabashedly schema- and XSLT-oriented, Site Manager is very much an HTML-oriented, page-centric system that demands little or nothing from users (of any level) in the way of XML knowledge, or managing content in a componentized, “chunked” way.

You can combine headers, footers, and boilerplate items into “Styles,” which are basically skins (the unchanging parts of a page that wrap templated content). Styles are easy to create and modify, and can contain nested components.

Both Template formatting and Styles consist of markup with <t4> tags as placeholders for content fields and navigational objects (menus, breadcrumbs, etc.). These are parsed dynamically on the application server, and a copy of the assembled code is pushed out to the web server. There is a nice, forms-based interface for adding such tags, and there is hope for the non-technical power user to manage content types and some degree of formatting in this system. To create your own custom navigational tags, however, you’ll need a Java programmer.

Performance

Since files are pushed to the delivery environment, web server performance is independent of the rest of the system and will be stable when content is predominately static. Publishing the content is another matter; it becomes mostly a load on the production server, and therefore is often a scheduled job (often nightly). If you need to push more frequently, be sure to test before signing the dotted line. TERMINALFOUR says it’s working on multi-threaded publishing (one thread per language) and other publisher optimizations.

On the back-end side, the system now supports clustered servers. In the TERMINALFOUR infrastructure, this means that editors can be supported by servers on their own continent. The company says some customers employ this geographic clustering, but we were unable to speak to them. You will want to verify its efficacy yourself.

Content

Contributor Experience

TERMINALFOUR prides itself on its acute awareness of accessibility issues. The system’s user interface is clean and consistent, but in the past, it has favored cosmetics over clarity. Starting with version 7.1, the product has colored icons and a distinct toolbar on top, which have made the interface clearer. It’s more in the style favored by WordPress and Joomla. (TERMINALFOUR could have picked worse style guides; both systems are favored by users for their usability.)

That said, the user interface is click heavy, without AJAX features like drag and drop. This

Content	
Contributor Experience	
Overall Usability	
UI Accessibility	
Contributing Content	
Authoring & Transformation	
Tagging & Taxonomy	
Content Reuse	
Media & Document Management	
Repository Services	
Content Lifecycle	
Workflow	
Globalization	
Archiving & Compliance	

could prove annoying, although customers migrating to Site Manager tell us it's an improvement over previous manual approaches.

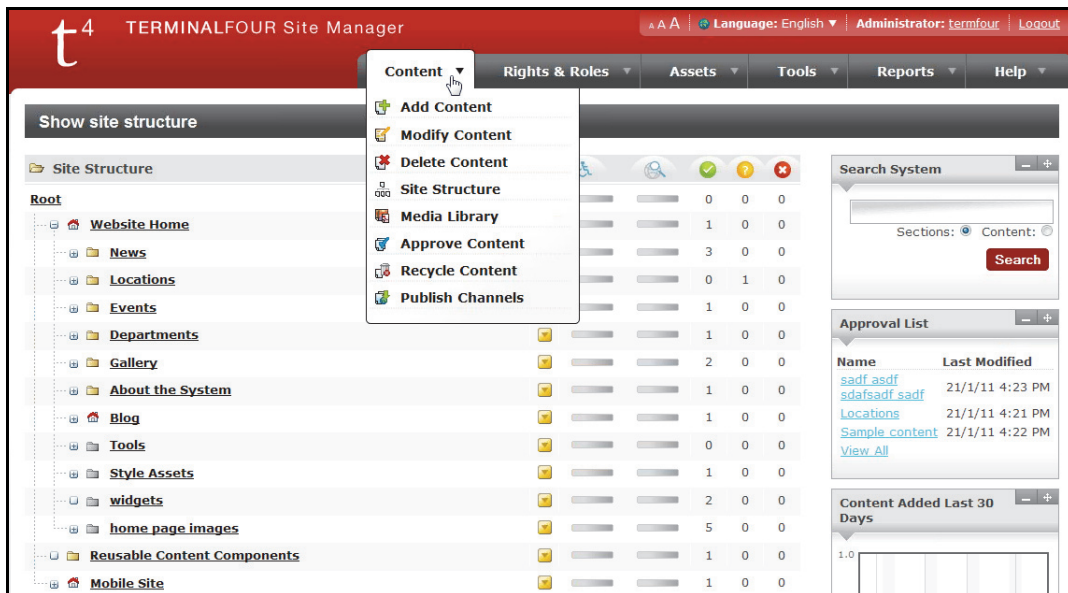


Figure 434. TERMINALFOUR version 7's interface is clearer and easier to see at-a-glance than previous versions were. Note the customizable widgets on the right, which users can willfully add or rearrange.

Users can add widgets to their back-end, similar to Ektron or EPiServer. Some of these are TERMINALFOUR, e.g., an “approval list,” or repository search. The vendor claims you can also add any W3C-compliant or Google Gadget widget to have a quick look at your analytics.

Contributors can be allowed to work in a few sections of the sites or specific types of content, as circumscribed by management. By default, the software will let contributors see only the areas and content types that they are entitled to see (which is usually a good thing).

The user interface has been internationalized (so that it can be localized), with localizations for French, Spanish, German and — interestingly — Welsh. The product ships with English and Welsh so you can check out the localization. Presumably, partners will localize the interface for other countries where TERMINALFOUR has been active, such as Poland.

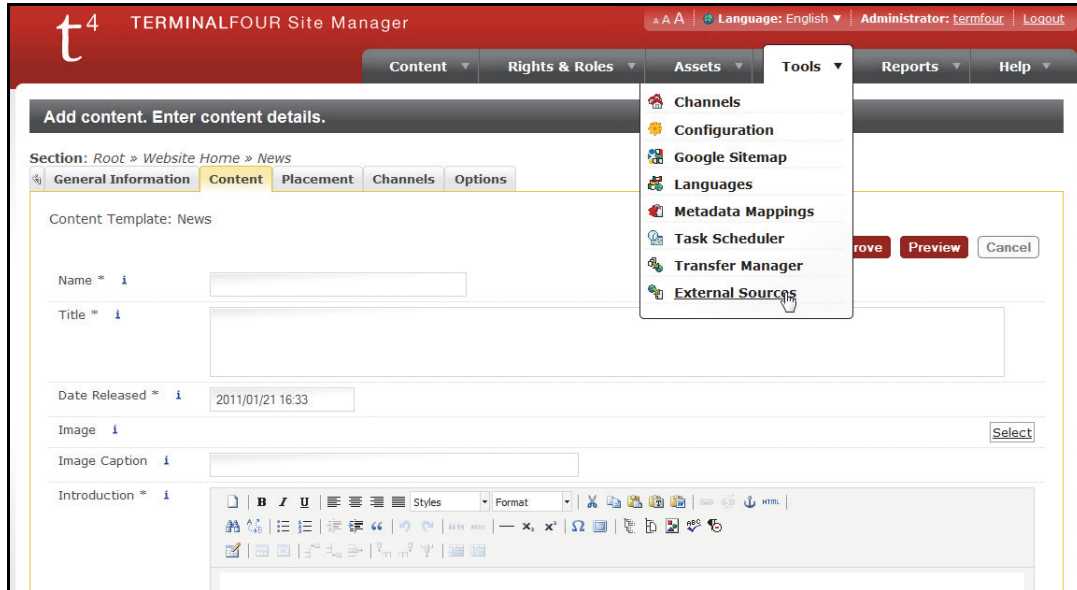


Figure 435. TERMINALFOUR uses the standard TinyMCE editor for rich text fields.

Contributing Content

Like most other systems, Site Manager provides the option of either working with content through the administrative interface featuring a WYSIWYG editing environment, or working in-context directly on the web page. The former option is “forms-based editing,” and the latter TERMINALFOUR calls “Direct Edit.” Starting with version 7.1, the rich text editor is TinyMCE.

Site Manager can support multiple taxonomies, but the absence of drag-and-drop support may make it a bit clumsy to manage hierarchical vocabularies on an ongoing basis.

By default, the product organizes content according to your site tree. Indeed, on the whole Site Manager feels like a very page-based system, thereby reflecting its web focus (as opposed to a DM, DAM, or general-purpose CM focus). Its customers probably find this to be more of a benefit rather than a demerit. For example, universities typically don’t have a strong need for sophisticated component reuse. You can create a kind of common pool of content in a part of your tree that never is published, but there is no strong sense of “placeless” content here. Moreover, even basic page repurposing across different locales can be challenging.

This is not an optimal solution for organizations that want to repurpose or reuse content across different site segments or locales. Content in TERMINALFOUR is tied to a content type and an individual section on the site, thus making content reuse across the site more difficult. An individual content item can be “mirrored” (linked to original content), as well as duplicated (create a second copy) to a different section in the same site or other microsites in the TERMINALFOUR root. This may not prove to be a viable option when working with multiple sites that share content. In short, Terminal Four is fundamentally a page-oriented product.

You can upload and manage various media in TERMINALFOUR, however customers find the process a bit cumbersome. Alternatively, customers can use WebDAV. Binaries have

associated metadata (such as category tags). Usefully, images can be stored in several “variations,” for instance, different dimensions. You can crop the image from the interface and store it as a new variation (though there are no controls for contrast, color or orientation). It’s not exactly DAM — not even DAM “Lite” — but it will suffice for many scenarios.

TERMINALFOUR can connect to WebDAV or CMIS sources and add them to the interface as “resource libraries,” which means editors can easily copy content from these external systems. CMIS, especially, is all the rage right now, and quite a few Document Management systems (including Microsoft SharePoint) support the standard for sharing their content, but so far, few web content management systems can be a CMIS “client” to use the connection. Note, however, that where WebDAV has quite a few cross-compatibility issues, CMIS likely has even more; test this functionality carefully with the system you want to connect to, and don’t assume that it will work.

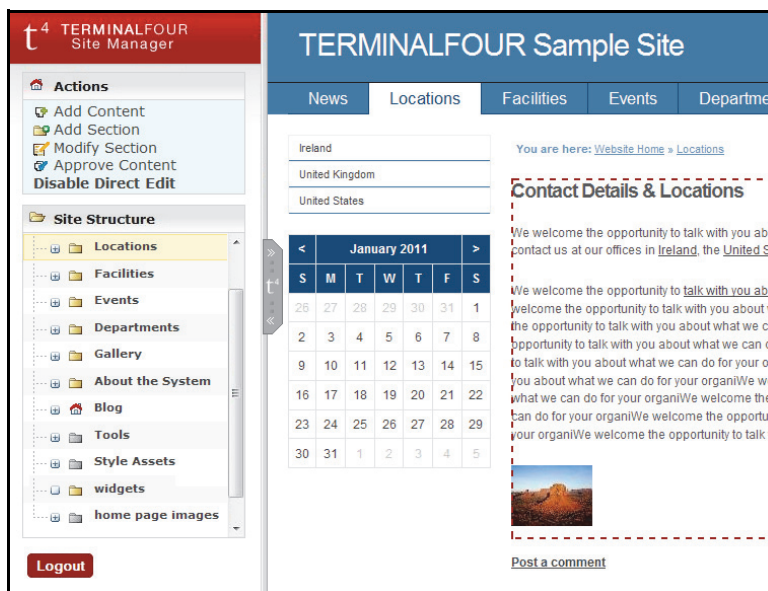


Figure 436. TERMINALFOUR’s in-context editing is called “Direct Edit.” Editable areas are marked with a dotted line; to the left is a simplified UI pane that can be collapsed. Editing is through a forms-based editor, not in-line.

Content Lifecycle

You create workflows using a web-based wizard. You select the number of steps, and then designate logic to apply to the steps. Each step of a workflow, for example, has a “Reject to” configuration option, such that any given step can be set up to do one of five things upon rejection by the approver:

- Return the item to the owner
- Return it to the previous step of the flow
- Return it to an arbitrary step (“Step X”)
- Send it to a new workflow altogether
- Do nothing

A variety of time-out functions exist as well, and administrators can short-circuit workflows to publish content instantly, using a variety of “fast track” options. The availability of these kinds of features in combination with fan-outs and quorum-based processes is somewhat unusual at this tier, and may account for some of the product’s success in the markets it serves.

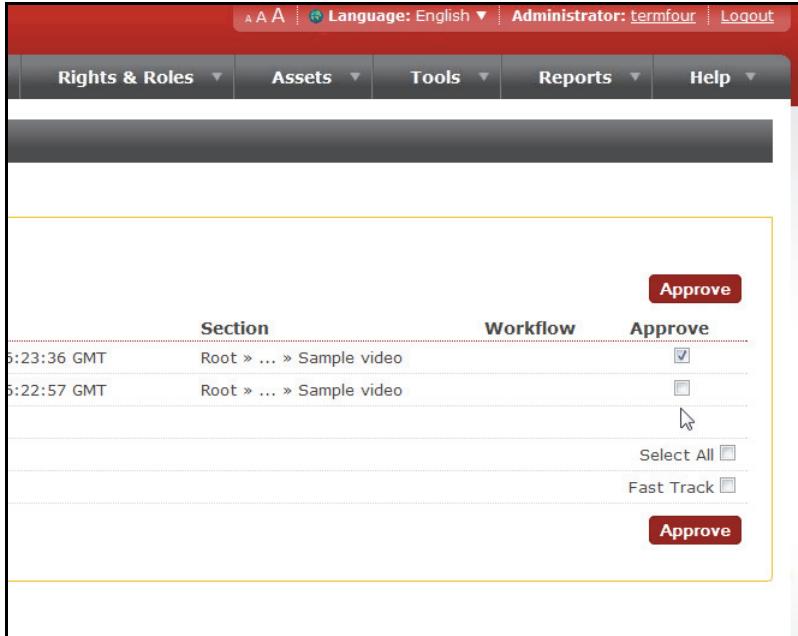


Figure 437. Approving content in a TERMINALFOUR workflow.

The system uses file locking and check-in/check-out functionality for version control. Version compare shows the differences between two versions, with the changes marked in green (added) or red (deleted). Customers tell us that this can be a bit challenging at times to handle; unlike the traditional versioning, it often results in unwanted multiple versions of the documents showing up in the author’s console, which can be confusing.

Wizards make it comparatively easy to deploy new websites in Site Manager. Without the need for extensive development, most web designers will be able to create required styles and other artifacts to roll out microsities quickly.

Content can be entered in multiple languages, and Site Manager is used for many public, bilingual sites in Ireland. Languages can be mirrored; new sections and pages are automatically added to other languages when they are created in another language. However, these will remain empty until they are edited and will not trigger a localization workflow; custom reports have to be set up to notify editors of a translation job at hand. Similarly, changes in the content do not trigger actions on their peers in other languages unless specific reporting is used. In short, this product is comparatively weaker here than, say, Sitecore or other European tools in upper tiers. In short, the product has some promise for globalization scenarios, but likely will not suit customers that have frequent changes to localized content.

Experience

Publishing

With its focus on government and higher education, it's not surprising that TERMINALFOUR produces relatively clean code and most sites conform to W3C and accessibility guidelines.

Digital Marketing

TERMINALFOUR doesn't provide analytics; most customers use Google Analytics or other third-party products.

Experience	
Publishing	
Standards Adherence	●
Multichannel	●
Mobile	●
Digital Marketing	
Site & Campaign Analytics	●
Testing & Optimization	○
Segmentation & Personalization	○
Social Media Integration	●
Promotional Campaigns	○
Community & UCG	●
Workplace	
Collaboration & Networking	●
Dashboard	○
Ancillary	
Site Search	●
Online Forms	●
Module Ecosystem	○

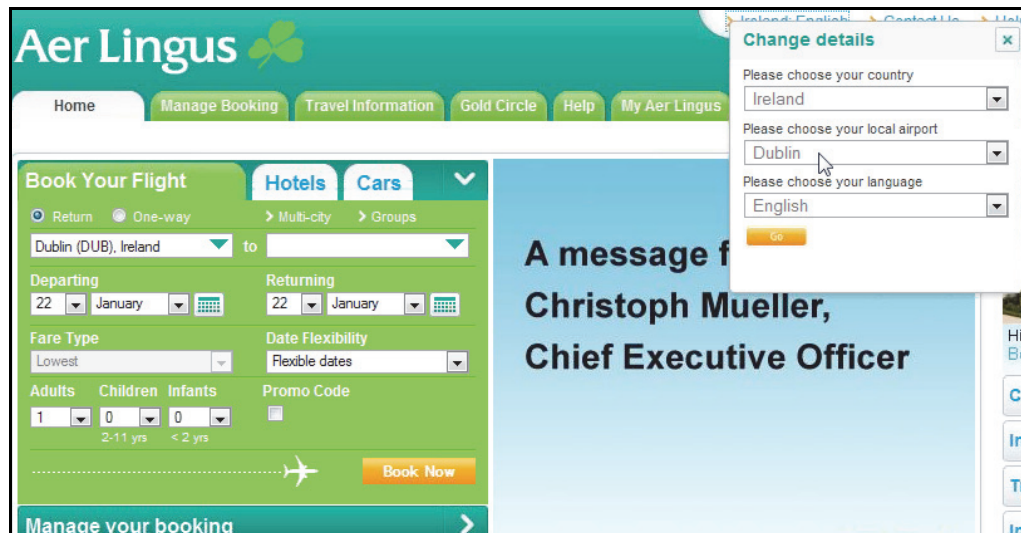


Figure 438. Example of simple personalization: the Aer Lingus website allows visitors to set their preferences for country, airport, and language, and it displays different content based on these preferences.

Ancillary Services

TERMINALFOUR offers licensees Exalead as an extra-cost option for search. The vendor offers some micro-applications via an “Intranet Pack” that isn’t productized but can be installed for you by the company’s ample services team. The software is actually included free with every Site Manager sale, but few customers seem to be able or willing to set it up on their own. The Pack includes discussion forums, calendaring, polls, a people-finder app, lightweight reservations systems (meeting-room booking), and the like.

The system allows editors to drop widgets into pages in the website (as well as in the interface they use themselves). The widgets can be Netvibes, iGoogle gadgets, or any W3C-compliant widget. TERMINALFOUR calls this a “portal-less portal,” since it enables customers to create what most expect to get from a portal — without having to install, or use complicated portal software. Of course, you shouldn’t expect it to be able to integrate anything more complex than these shallow presentation blocks; for that, you will need an actual portal system. However, it’s an easy way to add a YouTube video or Flickr gallery.

Vendor Intangibles

In its headquarters in Dublin, Ireland, TERMINALFOUR has 60+ employees. The largest chunk of TERMINALFOUR’s revenue comes from the United Kingdom and Ireland. The vendor has opened an office in Boston and has had quite a few wins in North America, mostly in higher education. The company says the US now accounts for about 38 percent of its revenues.

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

In 2009, the company took on outside investment through equity financing (previously, it was wholly owned by the founder), to fund further geographic expansion and product development. TERMINALFOUR doesn’t seem to be squandering this modest infusion. The first and most obvious result is the overhauled version 7, and the money has enabled the company to hire new staff.

Though TERMINALFOUR says it does see business from other areas (finance and retail), its main traction still seems to be in government and higher education. It says its product now is used by 30 percent of the top 50 UK Universities. Many of its technological improvements have been targeted at this market, which is why it’s not surprising that the system scores well against the scenarios we commonly see there (such as microsites and ultra-large single sites).

Some projects are implemented with partners, but most are completed by TERMINALFOUR consultants. Customers often let TERMINALFOUR take care of the upgrades, as well. Be aware that outside of the UK and Ireland, experienced consultants are rare. The fact that T4 still takes on so much services work has probably been a good thing for the product overall, since it has given the company direct exposure to day-to-day customer pain — but it also means that you probably can’t turn to your favorite integrator for support. The company says it’s building up its partner network beyond Ireland and the UK. Some of TERMINALFOUR’s

Irish members have moved to the US, and TERMINALFOUR says it is building out the team there.

What TERMINALFOUR lacks in consulting partners it seems to make up in customer networks, at least among universities. At the end of each year, a customer event is organized; at the end of 2010, some 150 customers attended.

TERMINALFOUR supports an extranet to hold the Site Manager's help repository with documentation, release notes, manuals, tutorials, and other helpful resources. Unfortunately, not all information is updated; it's not uncommon to see documentation updates lag behind almost six months after product releases. Also, the extranet contains multiple copies of the same documents, which can be confusing.

Over recent years, TERMINALFOUR has ramped up its release schedule. Minor releases now come about every 2 or 3 months. In order to appease customers with this increased release schedule, version numbering is minor.

The basic license cost is approximately €15K for a site with less than 7,500 content items. For an unlimited license, you'll pay €50K. These costs are independent of custom configuration and setup, which are billed on a time and materials basis. TERMINALFOUR says customers now can opt for a service called "Sure Start," a fixed-price implementation for simpler projects. Annual support comes in comparatively high at 22 percent, but there are no additional charges per server, per user, or per site.

Conclusion

TERMINALFOUR has the advantage of a well-developed home market, where they perform quite well, and have maintained a reputation of good support. Outside of Ireland and the UK, the vendor is mostly focused on the eastern US. The company has a few customers elsewhere, but these seem to have been mostly lucky coincidences.

Accessibility of the contributor UI is a strength, and a major reason for wins in education, government, and other public organizations. Site Manager enables rapid development of new sites and quick rollout with decent user acceptance. It offers a flexible and relatively easily configured workflow system that seems to have found a sweet spot with customers. Like any Java system, Site Manager will require some custom code for those who dare to extend the platform, but at least the API (for the entire product) is open.

T4's comparatively inexpensive — if unusual — licensing model benefits those with small amounts of content. However, those with more content must pay dearly for it. As to the company itself, there is a very collegial informality that some customers may welcome and others may find frustrating. This is not an adrenaline-fueled California startup. Although it is professionalizing, it's still run like a family business, but the impression is that TERMINALFOUR will try to take good care of you.

It appears that TERMINALFOUR is entering a new phase of its existence. After a long period of relatively slow, steady growth, company revenues are increasing sharply (along with head count). Rapid growth can be destabilizing, however, so it will be worth watching carefully to see how well this small company manages to keep its grip on things like quality control and customer service as it heads for the next bend in the roadmap. This becomes comparatively more important with TERMINALFOUR than with other vendors, since there is a good chance

that you'll work directly with the vendor (and not a third party) during the actual implementation.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Automattic, Inc.: WordPress

wordpress.com

Project at a Glance

Specsheet	Automattic, Inc.: WordPress 3.9 Summary
Geography	Global
What's New	<ul style="list-style-type: none"> • New, upscale hosting and support options from Automattic and others • Roadmap published for major 4.0 release • Noticeable uptick in media customers
Strengths	<ul style="list-style-type: none"> • Well suited to Simpler scenarios • Broad and deep open source community, which has localized the platform into more than 70 languages • Unusually simple product to install • One of the few products that is almost universally liked by all contributors • Many third-party modules available • Extraordinarily diverse hosting/cloud options for nearly any budget • Growing traction among media firms • Excellent fit as "second" CMS for enterprise microsites
Weaknesses	<ul style="list-style-type: none"> • Rudimentary user management and limited access control reduces suitability in many enterprise environments • Has no native content modeling: still requires development or plugin for anything beyond "page" and "post" types of content • Many key enterprise features are not in the core, and must be realized through third-party modules of variable provenance • No at-a-glance view of the page hierarchy and lack of in-context editing limits manageability of larger sites • Irregular new releases in quick succession have administrators tied up in a continuous update process • Popularity and openness of the software leaves sites persistently vulnerable to malicious attacks • Comparatively more difficult to integrate with other enterprise systems
Potential Fit	Informational, Microsites & Landing Pages
Unlikely Fit	Complex Scenarios
Compare To	Joomla, Drupal, DotNetNuke, TERMINALFOUR
Operating Systems	Windows or Linux
Repository	MySQL
Client	Browser
App Platform	PHP

Specsheet	Automattic, Inc.: WordPress 3.9 Summary
Licensing	Open Source (GNU GPL)
Ownership	Privately held with 250 employees

Summary

Currently the most popular system for blogging around the world, WordPress is universally admired for its simple setup and its ease of use. This has caused the system to gain additional popularity as a more traditional web content management system. For simple scenarios, you could do quite well with the software — which, in this case, could prove to be more apt than Drupal or DotNetNuke.

However, you should understand the long history of WordPress as a blogging tool; used as a CMS, you could run into its limitations quickly if you need more enterprise-y features like workflow, content reuse, and ERP integration.

The latest versions of the product have improved its usability, but WordPress remains very self-contained. For most enterprises, WordPress will best suit as a blogging or microsite platform — complementing more complicated and enterprise-oriented Web CMS tools.

A possible exception is within the media world, where you can purchase some attractive (if expensive) hosting and support options from WordPress’s parent company Automattic, among others.

In sum, the platform has become more scalable (in no small part by being eminently “cloudable”), but its continued emphasis on simplicity comes at the expense of a narrower set of functionalities and fewer customization hooks than competing products.

Scenario Fits	
Simpler Site	
Informational	<input checked="" type="radio"/>
Microsites & Landing	<input checked="" type="radio"/>
Mid-Range	
Basic Digital Marketing	<input type="radio"/>
Mobile Site	<input type="radio"/>
Community Oriented	<input type="radio"/>
Complex Site	
Advanced Marketing Portal	<input type="radio"/>
Global Enterprise	<input type="radio"/>
Multichannel Publishing	<input type="radio"/>
Ultra-Large Single	<input type="radio"/>
Digital Workplace	
SMB/Departmental	<input type="radio"/>
Global/Enterprise	<input type="radio"/>

Introduction

The product started out in 2001 as “b2/cafelog,” a PHP/MySQL tool described as “logware,” and it was bare-bones blogging as straightforward as it gets. Feature requests and the start of a fresh codebase led to the 2003 inception of the more advanced product called WordPress.

In a somewhat unusual turn of events, in 2005, several core members of the WordPress open source project started their own company, Automattic. They built a free, hosted service, wordpress.com, as well as version 2 of the installed product.

While Automattic is generally seen as the company behind WordPress, it might be more accurate to see Automattic as WordPress’ company, which has led to some confusion. The software project (wordpress.org) is more or less led by Automattic as a “benevolent

dictatorship,” but it is open source nevertheless (under the GNU GPL license), with an active and supportive community. Over time, Automattic has released a bevy of different commercial offerings, and a broad ecosystem of hosting and support companies have also emerged. In addition to the open source WordPress codebase that you can download and install yourself, Automattic now has six different packaged offerings as displayed in the chart below.

Flavor	Hosting	Support	What You Get	Cost
.org	Self-Hosted	Community	Source code to the core software (and themes) that you can install and host yourself	Free
.com Basic	Automattic hosted	Community	Automattic hosts and provides some basic services stats, comments, polls, social integration, 3 GBs of storage, along with a choice of 150+ themes, but you have an xyz.wordpress.com domain name, and your site has to host their ads. You can pay small fees annually for niceties: enhanced designs, custom URLs, videos, no ads, and so on.	Free, but Automattic will put ads on your site and may limit the type of content you publish
.com Premium	Automattic hosted	email	Custom domain (no WordPress in the URL), ability to upload videos, 13 GBs of storage, no ads, custom design tools (font and CSS editing), and some commercial support. Here and below, Automattic provides security and backup	\$99/Year
.com Business	Automattic hosted	Chat	All premium themes, no ads, unlimited storage, live chat support during US business days, custom domain, custom design, the ability to upload and embed videos, and the ability to conduct e-commerce via integration with Ecwid, Shopify, or ShopLocket	\$299/Year
.com Enterprise	Automattic hosted	Unlimited Tickets	Unlimited traffic and storage: about 100 WordPress plugins, all WordPress.com themes, customization tools and a JavaScript editor, and a more enterprise-y support structure. However, you still can't do multi-blogs, custom themes, CSS, or plugins	\$500/Month

Flavor	Hosting	Support	What You Get	Cost
VIP Saas	Automattic hosted	Unlimited VIP Tickets	Unlimited traffic and storage with advanced support (fees vary here), CDN, basic video services, all of the main plugins, and spam protection. Beyond that, you can purchase a bevy of fee-based, add-on services, like monitoring, liveblogging, better search, and video. You can also pay Automattic consultancy fees (ranging from \$15K-115K+) to migrate your existing site to the platform.	\$5,000/Month for hosting and basic support for one developer. Additional seats and/or faster support can run an additional \$2K–20K per month per seat. Lucene-based Elasticsearch runs \$1,100/month; liveblogging is \$500/month
VIP Support	Self-Hosted	Ticket-based	Ongoing support per year per seat, with a response time of 4 hours to one day, which determine a very wide range in fees. Support is for current and one previous version of WordPress only.	\$15–\$250K per year, per support seat. Onsite training courses cost \$15–\$30K/day. Architectural and security reviews run \$60–75K. Other reviews and monitoring somewhat less.

Today, Wordpress.com is home to more than 13 million sites, and we estimate more than 90% of them are blogs. Because of its popularity, some organizations choose to run their small business or corporate brochure websites on modified WordPress templates, as well. Moreover, an increasing number of larger enterprises employ WordPress as a microsite platform. It's relatively easy to change the templates to be less blog-like, pick and choose from the plethora of plugins, and use pages (instead of "posts").

This has led the project to describe the software as a "publishing platform," that has "evolved to be used as a full content management system." However, you'll see that the interface is geared toward blogging first, and more traditional web content management second.

WordPress now powers blogs for a long list of renowned names, including GM Next, Xerox, Arena Magazine, People Magazine, Green Bay Packers, Reuters, and VH1, as well as celebrities such as Usain Bolt, Rosie O'Donnell, Stephen Fry, and Martha Stewart.

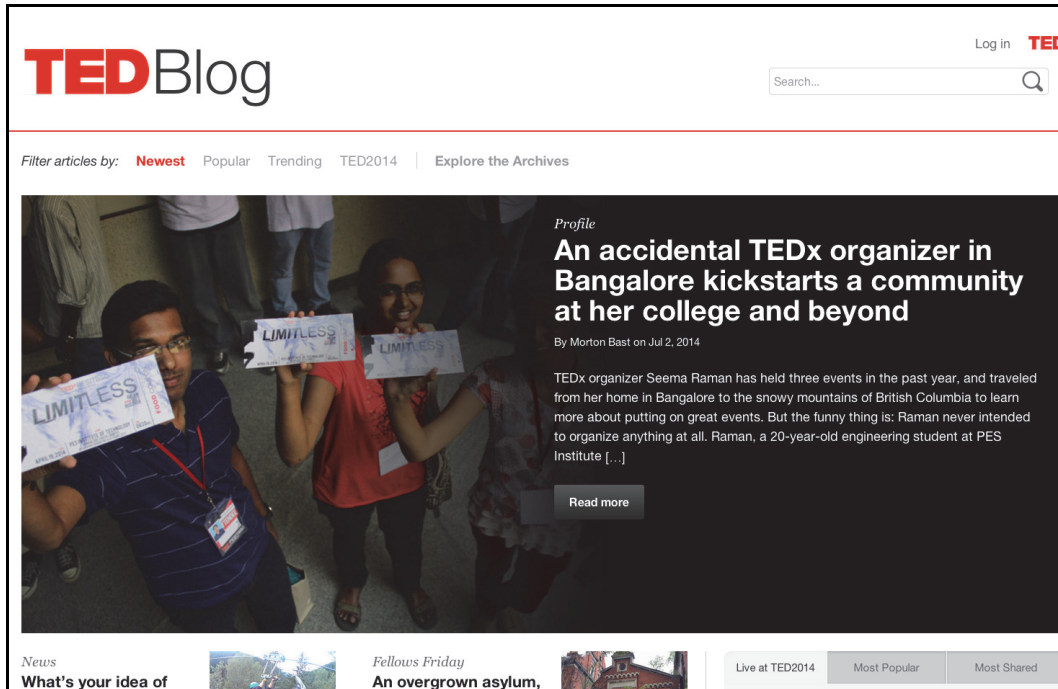


Figure 439. The TED Blog is run on WordPress VIP.

However, the list of sites where the software is used as a web content management system — rather than just a blog — is somewhat less impressive. It's used in academia, but mostly by departments (such as the University of Maine, Leeds City College, UC Berkeley Department of Spanish and Portuguese, and the Department of Music at Cornell). More often it's used for brochure websites (such as the Vancouver Convention Centre). Perhaps the most impressive examples of WordPress as a CMS to power a full-blown website are the Brazilian Culture Ministry, WSJ (“The Magazine from The Wall Street Journal”), and The New York Times.

The uptake within the media segment makes sense. You could consider WordPress as a kind of article accumulation machine. Automattic has ridden this wave and come up with some (rather pricey) all-you-can-eat hosting and support options that target this use case.

Version 3.9 was released in April 2014. (See “Vendor Intangibles” on page 830 for the status of Version 4.0, which is planned for release in Q4 2014.) It mainly featured rich text and image editing improvements after the significant December 2013 release of version 3.8, which focused on usability enhancements.

Technology

Core Architecture

WordPress is built on the PHP/MySQL platform, and will run on most web servers (including Apache and Microsoft IIS versions) with PHP version 5.2.4 or later installed, using MySQL version 5.0 and up. The product boasts a straightforward “famous 5-minute install,” and indeed, if the web server and database are already up and running, most of those five minutes will usually be taken up by uploading the scripts to the server, with minimal additional configuration.

On the other hand, upgrading to new versions (which often follow in quick succession) can be a bit of an undertaking. Admins responsible for maintaining blogs will have to set aside a fair share of time for unscheduled maintenance; being a point release behind is never a good idea with a popular open source platform (vulnerabilities are sure to be found — and exploited — then quickly fixed), but planned release dates are seldom met. You’ll want to install the update as it becomes available, but won’t be sure when exactly it’ll be released. This is a particularly important point for WordPress, since it is a hugely popular platform (so exploits are worthwhile), and it is easily identifiable by the version number embedded in the HTML source code (in the generator metadata).

Technology	
Technical Administration & Security	
Threat Prevalence	○
Authentication & Authorization	○
System Reporting	○
Multisite Management	○
Cloud Services	●
Development	
Configuration & Customization	○
Integration & Extension	○
Content Modeling	○
Templating	○
Performance	
Back-end Performance	○
Site Caching & Delivery	○

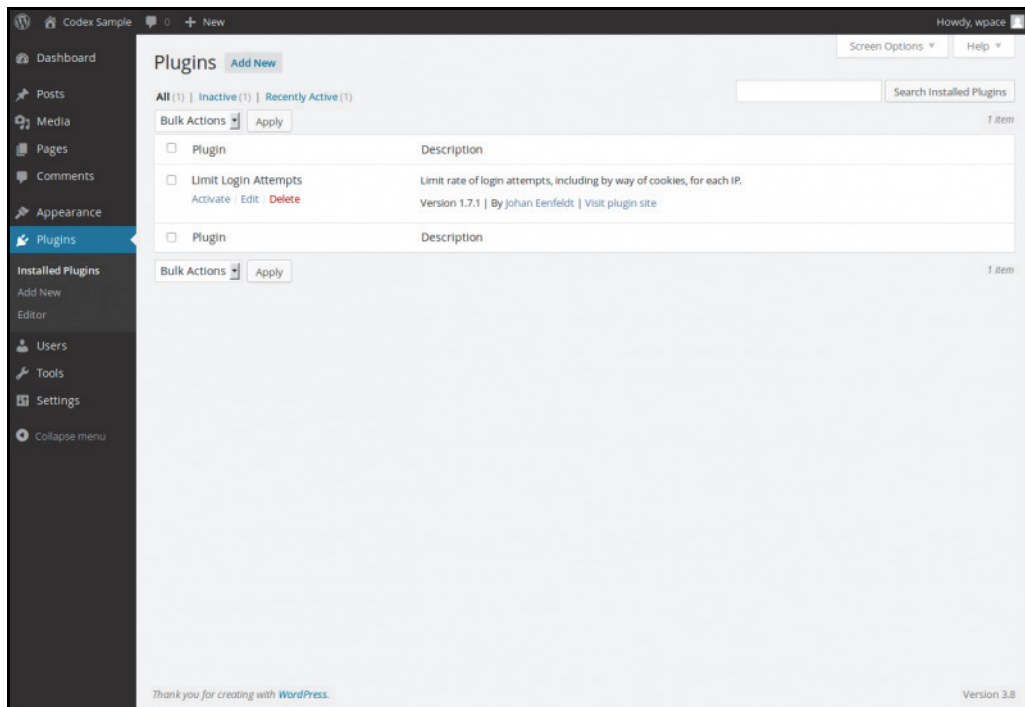


Figure 440. Administrative tools: managing plugins. Source: WordPress.org.

WordPress aficionados will counter by noting the one-click upgrade for the core software, not just the plugins. While you should be careful to back up your software and database before you proceed, upgrading is painless and quick. However, this does mean that you have to have your initial configuration right; restrict the access rights to the filesystem too much and the upgrade will fail — open them up too much and your installation will be vulnerable to attack. If you plan to install the software on your own servers, you'll need an experienced administrator, and it can be even more difficult to get right if you're relying on a hosted server. It will pay off however, as the next minor release will be as easy as one click; if the install fails, WordPress will even roll back to the previous (working) version. This process has become more reliable over time; it's now possible to upgrade multiple plugins at once.

As with most of the PHP/MySQL systems, WordPress delivers web pages dynamically from the database without much ado. The back-end interface draws from the same database, and is actually more heavy weight than the presentation layer. There is no option for static publishing; however, there doesn't really seem a reason for this option.

The larger point here is that WordPress (like Drupal), is meant to run as a self-contained system. That is, with the exception of identity and email systems, WordPress does not typically integrate with other enterprise services in the delivery tier. For example, it can incorporate services like e-commerce via plugins, but you will have to work hard to weave WordPress into your e-commerce platform.

Multisite Management

In versions 3.x, WordPress supports running multiple blogs off of one instance, since the former "WordPress MultiUser" (MU) has been integrated into the core product. However, setting it up certainly is not as easy as the regular "5-minute install." It will involve some additional steps, because by default, the feature is turned off. However, running WP in this mode is now relatively easy; the system will even generate the required .htaccess rewrite rules to configure your web server. (You'll have to add those rules to the server yourself though — which may be daunting to most WordPress users, but should be trivial for any administrator.) This makes creating a new site with use of existing plugins and themes very easy. However, WordPress assumes that each site will be a sub-domain or a directory on the same server — there's some trickery involved in getting sites to work on their own domain name.

Authentication and Authorization

With respect to user entitlements, WordPress has a straightforward directory, which displays as a flat list. The system only has six default roles: Super Admin, Administrator, Editor, Author, Contributor, and Subscriber. Super Admin is new; that person can manage all of the WordPress sites running off that instance. The Administrator has access to all rights within a particular site. Subscribers can read "password protected" posts and pages on the site (but have no other rights). The differences between Editor, Author, and Contributor are the ability to add pages (all three can add posts), modify existing pages, and delete. Contributors don't have the right to publish the posts they add themselves, and are therefore particularly well suited to guest bloggers.

Note that you can't set different permissions for site segments, folders, or other criteria. Any permission will be site-wide. This doesn't necessarily have to be a problem, but could limit a site with multiple branches and a large group of users. You may have to cut up the site in different WordPress installs to circumvent the lack of granular controls.

It's possible to use an external (LDAP) directory for users through a plugin. However, users report that it's difficult to get this to work. It's unlikely that you would use WordPress with many users, anyway; adding them by hand shouldn't be a great chore. This changes if you start making active use of the "password protected" pages; "Subscribers" will be in the same user list as the actual users of the system. Thankfully, you can filter the list for specific user types (and there's also a user search).

Reporting

The system doesn't offer any kind of logging or audit trail, other than some recent activity widgets in its dashboard. There are several plugins that address this deficiency, but as with any plugins, their quality varies; there's no support and there are no guarantees.

Cloud Services

WordPress can arguably boast the most diverse set of cloud-based deployment options of any system in this report. Your choices range from:

- Multitenant SaaS provided by Automattic or other firms
- Managed hosting of dedicated instances by various third-party vendors in various public clouds or commercial data centers around the world
- Managed hosting by AWS and Rackspace
- Self-hosting in AWS and other cloud providers
- Integrations with CDNs by most of these providers

As always, you'll want to investigate who's responsible for what carefully. In addition, just because you're in the cloud doesn't mean you have complete redundancy or wide-ranging geographic syncing. Note, for example, that Automattic's servers are only located in three Midwestern USA data centers.

For managed services, you may have to deploy a particular build of WordPress and limit the variety of plugins you can use. In some cases, you can deploy custom plugins, but the provider needs to test them before pushing them into production. Why is this? Read below about Threat Prevalence.

Threat Prevalence

WordPress exploits pay off for hackers. Find a hole in WordPress, and — due to the popularity of the software — you'll have an enormous installed base to which you can spread your links. Over the past few years, these attacks have grown increasingly sophisticated, and most of the modifications are invisible on a regular browser. Thus, many site and blog owners find out about the hacks only after Google suddenly bans their site from search results because of link spam.

This problem is compounded by the fact that most WordPress implementers remain far behind the current version. We estimate that at any given time, only 20 percent of WordPress customers use one of the two latest dot-releases, and the situation with plugins is even more fraught. In fairness, this is common with other CMS tools, but it makes WordPress a target-rich environment of unpatched vulnerabilities.

In 2010, a large number of WordPress installations hosted by GoDaddy was affected by such an exploit, presumably because the host didn't allow the fine-grained security settings that were needed to keep WordPress running safely.

In 2012 (according to various reports from security firms like Symantec and Kaspersky Labs), an expansive Flashback botnet of Mac computers was triggered by anywhere from 30,000 to 100,000 infected WordPress sites. This incident spiraled into hundreds of thousands of virus-laden Macs.

Today, WordPress remains a huge target that only gets bigger. Hosting services will often shield you from threats by managing patches and vetting plugins themselves, but you still need to remain vigilant.

Ultimately, security is more of an organizational process than a software attribute. With WordPress, however, you need very solid processes.

Development

WordPress has only two content types: posts and pages. This harkens back to its origin as blog software, with pages later added on to provide some static info, such as the “about” page of a blog. In releases in the past three years, however, both have been expanded to the point that they're almost indistinguishable. The main difference now is that posts can be tagged and categorized. Additionally, pages can have a hierarchy (you can assign them a “parent” page). You should probably ignore the “page” and “post” monikers and use whichever is the most appropriate for your scenario:

- Using “posts,” you can build a site much like you could in Drupal, with emphasis on tagging and taxonomy-based presentation (this is what media firms tend to like)
- You can choose a more traditional organization, with “pages” in folders and a page tree

Note that there is no notion of true content modeling in WordPress — one of its biggest differences with Drupal. WordPress does have a notion of “custom post types.” When you create one of these, it creates a new section in the toolbar, similar to that for “regular” posts, in combination with “custom fields.” This means you can add your own content types, similar to Drupal's “custom content kit.” “Custom fields” can be used as metadata in your templates, but these are free-form key/value pairs; users must enter both (i.e., the key “color” and the value “yellow”). They can add as many as they like, but unless you give users some sort of a cheat sheet, they won't know what to enter. The help function in the edit interface links back to the development help — not to anything that's useful in describing what they're expected to enter, however. In short, expect no real flexibility here out of the box.

If you want to provide actual form fields, radio buttons, drop-down lists, or mandatory formats to your content, you'll have to extend the interface; this will take a PHP developer to create “custom write panels” (interface modifications). Alternatively, you can use one of the plugins (such as Custom Field Template, Flutter, or Magic Fields), which provides the content modeling that WordPress lacks. These actually will add new content types to the interface, and use custom fields for the repository storage in the background. Still, be aware that this will also make template development a lot harder. It's still possible, but not really what WordPress is particularly good at doing.

As a result, WordPress is not ideal for componentized chunking and content reuse more generally; the fact that it tends to bind content types to specific themes compounds the problem (see “Contributing Content” on page 826)

WordPress uses PHP as the backbone of its templating, embedded as calls to functions. Many pre-built templates are available, and you can perform basic code editing through the interface. There have been complaints, however, that the quality of the templates varies. To avoid the recognizable look of many WordPress sites, or to have the design fit in with a specific style, more than a keen eye and HTML/CSS skills will be required. You need knowledge of both PHP (the templates are coded in PHP), and the specific WordPress functions (called as PHP functions) to get a template to work. Be aware that WordPress’ internal functions aren’t as flexible as you’d see in many other systems in this report, and they usually suffer from annoying limitations — which you’ll have to overcome in PHP.

The software has a basic editor for template and plugin code built into the administrative UI. Most template developers still tend to prefer to open the files in an external editor. Because of the straightforward nature of the system, developers are tempted to work directly on the template files of the live site, saving them to see what the effect is as they go along. The system itself doesn’t provide any configuration management or versioning of templates. On the plus side, it’s quite easy to export and import content and templates.

Thus, configuration management could present a challenge. WordPress keeps all content and settings in the database, and all templates as PHP files in the templates directory. Make sure you’re working with a local install for test purposes, and then copy it to the live site as a new template when you’re done. With the flick of a button, your site will show the updated design (and with another flick of the button, you can switch back if something goes wrong). You could also take cues from wordpress.com’s “VIP hosting,” which requires that developers make changes by committing to an SVN.

The software can be extended with plugins. The directory currently lists some 32,000 of these add-ins, but since there is no official mechanism for testing the quality of the plugins, your mileage may vary. A common problem is that plugins ignore the prefixes in database table names. As a result, they don’t function correctly when multiple sites share the same database (i.e., when you enable the multisite capability). Careful testing is in order in those environments.

Tellingly, Automattic’s controlled environment at wordpress.com only lets you use 19 plugins. VIP extends this to about 100 plugins. WordPress also supports widgets, which are part of the front-end design of a site — and can be placed there by users, without the need for an administrator. In general, though, most users tend to prefer plugins (added to the back-end of WP) to widgets (added to the presentation side).

Performance

Since WordPress publishes dynamic content, this means the usual caveats to the “frying” process apply. Both contributors and visitors will draw on the same server resources, and if WordPress fails, the site also becomes inaccessible. For high-traffic sites, you may need to take additional caching measures, using standard industry approaches. There are caching plugins, but some users have reported that they may actually downgrade performance.

Still, the software is very lightweight, and even if you compare it to other, more complicated PHP-based systems like eZ Publish (which can cause major headaches to get to run smoothly);

WordPress comes out quite favorably in this respect. This also means that it's easy to use an inexpensive third-party host. Many high-traffic WordPress sites use CDNs like Akamai, although many others seem to prefer to scale up and out instead.

Content

Contributor Experience

The WordPress interface opens with the “Dashboard,” which attempts to give at-a-glance access to the main functions and recent activity (Figure 441). The interface seems to be well-liked by users, and recent usability improvements show a community quite committed to user-centered design.

Functions are accessed from a horizontal toolbar, which can be collapsed to an icons-only version to save screen real estate. The dashboard and post screens to the right of the toolbar can be re-arranged with simple drag-and-drop functionality.

Content	
Contributor Experience	
Overall Usability	●
UI Accessibility	●
Contributing Content	
Authoring & Transformation	●
Tagging & Taxonomy	●
Content Reuse	●
Media & Document Management	●
Repository Services	●
Content Lifecycle	
Workflow	●
Globalization	●
Archiving & Compliance	●

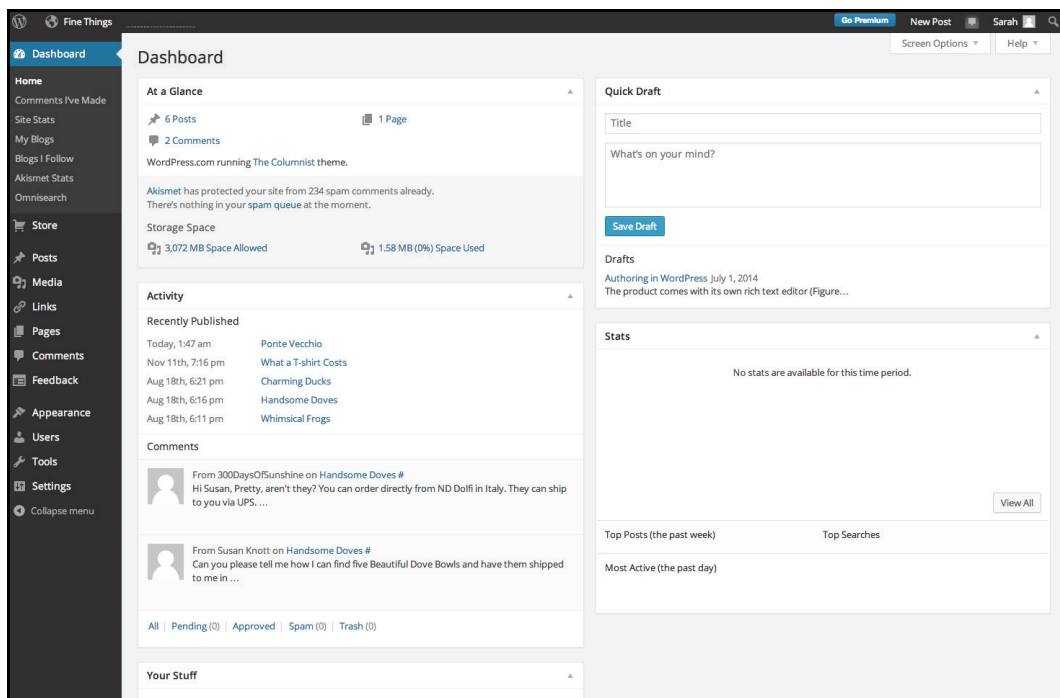


Figure 441. WordPress’s Dashboard.

The accessibility of the interface is fairly good; it makes sensible use of color to indicate active functionality, and recent improvements have provided better support for screen readers.

The interface has been localized into a quite impressive 70+ languages; you can download localized versions, or install language packs. However, most translations are maintained by volunteers, and it's not too surprising that many will lag behind on several versions. They will still work, but there may be missing labels, or employ occasional English in the mix.

Contributing Content

To add content to a WordPress site, you have to use the administrative back-end. That is, there is no in-context editing (where you browse an editorial instance of the site itself and make the changes there). You can understand its absence; in blogging and news environments, you tend to add content more than edit and modify it. However, the lack of in-context editing is increasingly lamented in the community, and is slated for a future major release.

The product comes with its own rich text editor, based on a simplified version of TinyMCE. Clicking a button called "Display Kitchen Sink" can bring out a second row of icons for less common functionality. Although it maintains a relatively minimalist look (even though the screens are certainly more crowded than in previous versions), small inline features add minor improvements to the experience. For instance, while typing, the editor will count the number of words (and continuously update the number).

There are two spell-checker options: TinyMCE Spellcheck and Jetpack, which now includes After the Deadline. The "Paste-from-Word" function seems to remove the dreaded Office HTML code, and the community continues to pay close attention here.

A nice feature is the ability to move around or collapse the various elements on screen. The "publish" dialog, can be dragged from its default, upper-right-hand position to below the edit form, and will resize nicely to the new width. The layout is easily rearranged, but it would have been nice if the admin could set defaults or lock the arrangement. As with the "Kitchen Sink" (which casual users often overlook or accidentally hide), authors may move elements unintentionally. This can be confusing.

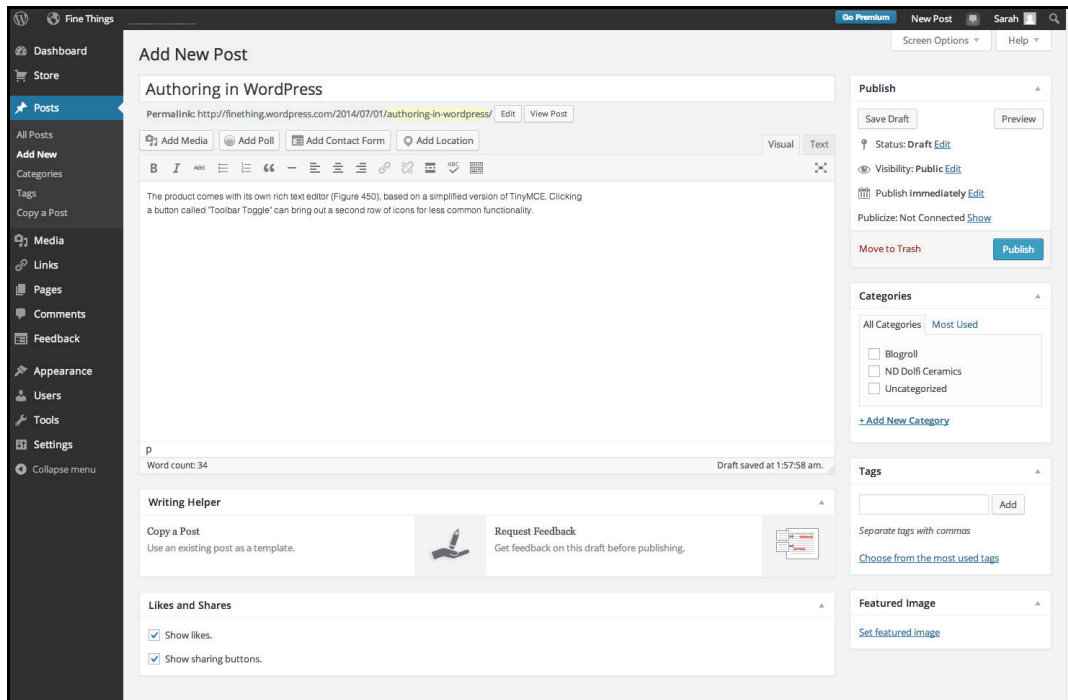


Figure 442. WordPress's editing screen.

If you use the “post” content type, or a “custom post type,” categories easily are added by checking the appropriate category box. In WordPress, the categories mostly are used to create site sections — and the UI reflects this. The list of check-boxes isn’t well-suited for large numbers of categories.

There’s also the option of using a folksonomy (uncontrolled vocabularies), which has its merits for more detailed “tagging,” and the templating engine can turn the tags into a visual tag cloud. Posts are more or less placeless. They appear anywhere where a template calls specific posts, based on categories or tags, similar to the way that Drupal functions.

The “page” content type, however, has no categorization or tags. Instead, you can define the page’s parent. This is a relatively crude way of defining where in the site tree the page will display — and it’ll become increasingly more difficult to understand your site structure as the hierarchy grows. You can also define the page template (which isn’t possible with posts). Pages are more traditional, web content management-like, but here, WordPress lacks relatively basic features, like showing a folder and page tree, so you can have some sense of the site structure. Not surprisingly, there is no real content reuse with pages, either; they’re one blob, in one spot. A community effort to build a plugin for more modular content was cancelled in 2013.

In short, you may find that if you need more than just the few static pages to support a blog, you’re probably better off taking the same approach as Drupal and DNN, using posts with the presentation basing its site structure on tagging, categories, and other metadata.

To insert an image, WordPress provides the option to choose an existing image or to upload a new one; the software will automatically create resized versions. You can also do some basic

image editing from the interface (i.e., cropping an image). Recent releases have improved image collection capabilities.

WordPress has limited repository search; for every list of content type (i.e. posts, pages, or media), there’s a search box, but this is limited to simple keyword search only. Don’t expect any of the fancy functionality that other systems offer — unless you license the Lucene-based Elasticsearch in WordPress’s VIP environment.

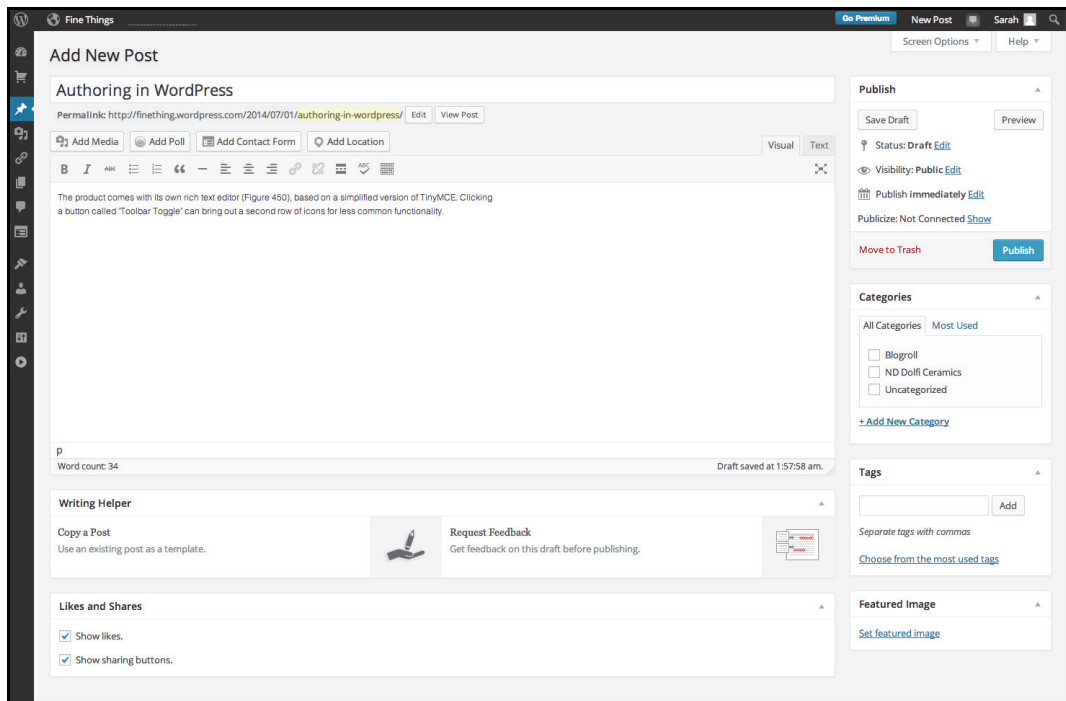


Figure 443. The edit screen, rearranged with the menu collapsed (icons only).

Content Lifecycle

Posts and pages can be saved, so you can continue to edit later; they will only be visible on the blog when you hit the “Publish” button. Of course, because of the dynamic nature of WordPress’ presentation, once something is published, all future edits (that were saved) will be live immediately. There’s no real notion of workflow, but you can designate some users as “contributors” and others as “editors,” and since the editors will be the only ones with publishing rights, they’ll have to approve the content. Effectively, this is as capable as the simple workflow functionality DNN has added in its latest versions.

WordPress can create and even clone sites from the basic admin interface. This makes it a breeze to create new sites. However, there is no notion of content sharing from the different sites; each site you manage is a separate instance.

Similar to the back-end interface, the front-end presentation elements of WordPress have been localized to many languages. There is no support for globalization; you’re assumed to have one site, in one language. Some sites work around this by using separate categories or tags for different languages. Of course, there are also plugins that address this, but their solutions aren’t very polished. If you want to publish in multiple languages in a more globalized (rather

than locally distributed) environment, you should probably look elsewhere (for instance, Plone, or TYPO3).

WordPress has versioning, which it calls “revisions.” You can restore previous versions, and there’s a visual compare that shows what changed between two versions. There’s a “trash” function; instead of deleting content, it’s now moved to “trash,” so an administrator can undo the deletion. The system is actually surprisingly capable in this respect, since several (sometimes very expensive) systems in this report don’t offer this kind of functionality.

A recent version has added an “Archive” state, where content can remain in the repository, but not get published to the site.

Experience

Publishing

By default, WordPress URLs use the cryptic syntax that is the hallmark of a dynamic delivery system (e.g., <http://mysite.com/wordpress/?p=437>). In practice, implementations use “permalinks” — permanent pointers to the blog posts. These can be completely configured to your liking, but it requires rewrite use. This creates intelligible URLs (<http://mysite.com/title-of-my-post>), or the date formats seen more often in blogs (<http://mysite.com/2015/03/01/title-of-my-post>).

The major issue here is that WordPress needs to be able to distinguish between categories and posts; this means that the URL of an individual post will always be in the yourdomain.com/category/title-of-my-post format. It’s impossible to remove the “category” part, because his could create a potential conflict between a category and a post name.

With “pages,” authors can define the page URL through permalinks. WordPress will determine if the URL already exists — as another page. In other words, it’s still possible to create a page with the name of a category, which could wreak havoc on a site.

WordPress produces clean, standard code, but you’ll need to mind the standards-compliance of any add-on module you employ.

RSS and Atom feeds can be generated for most of the streams in your site (such as posts, comments, etc.), and with specific plugins, it is also possible to ingest RSS from other sites in your blogs. Other than that, WordPress isn’t well suited to multichannel publishing.

In recent years, WordPress has improved its mobile publishing capabilities, most notably by remaking its default themes according to responsive design.

Experience	
Publishing	
Standards Adherence	
Multichannel	
Mobile	
Digital Marketing	
Site & Campaign Analytics	
Testing & Optimization	
Segmentation & Personalization	
Social Media Integration	
Promotional Campaigns	
Community & UCG	
Workplace	
Collaboration & Networking	
Dashboard	
Ancillary	
Site Search	
Online Forms	
Module Ecosystem	

Digital Marketing

Although the system has no analytics out of the box, several plugins cater to this. In practice, most seem to prefer third-party analytics, such as Google. Likewise, other e-marketing services can be found through a plethora of plugins. The quality and provenance of these will vary. Test carefully.

Social tools are only available through the myriad of third-party plugins and widgets, not the least of which are Automattic’s own commenting and spam-detection services.

Ancillary Services

Site search, like the repository search, is nothing fancy: keyword search only, with no advanced search. It’s quite effective, and it’s easy to template the search results page — although it’s quite hard to control what is searched. If you want more fully fledged search functionality, you should consider eZ Publish (with its Lucene Solr-based eZ Find), or license Elasticsearch via Automattic (for a pretty penny).

Given its pedigree in blogging, it’s no surprise that this is a particular strength of WordPress. The product shines in this respect, and support for commenting is very good, with all the fixings you’d want, including detailed controls and a fairly effective, free spam control service, Akismet. If your plans of a “social” or “community” website pivot on visitors commenting on posts and pages, WordPress should be on top of your list. Other systems have added commenting in recent years, but many of them still lack the controls they needed to turn it into a success. The Akismet plugin is free for non-commercial use. Commercial fees range from \$5/month for 80,000 checks for one site up to \$50/month for 80,000 checks per network for an unlimited number of sites. According to Automattic, 99 percent of comment spam is caught by Akismet (according to the Akismet Spam Zeitgeist, the percentage of checked comments that are spam is about 94 percent).

Vendor Intangibles

The open source WordPress project is clearly run by developers at Automattic. However, the community as a whole is large and thriving. A plethora of plugins and themes are available, with new additions daily. The ins and outs are well known, and in addition to the ample documentation on the WordPress site, there’s a sizeable quantity of walk-throughs, tips and tricks, videos, and several books.

Intangibles	
Vendor Services	
Vendor Professional Services	
Channel Partner Services	
Support & Community	
Strategy & Roadmap	
Viability & Stability	

Viability & Strategy

Automattic itself is a flashback to pre-boom hipness, with corporate job titles ranging from “quantum bug creator” to “happiness engineer.” The company has raised a whopping \$317M in venture financing over the years (most recently, \$160M in Series C funding in May 2014). However, this doesn’t appear to have changed them a lot, which we consider a positive. Presumably, a large portion of this funding is goes to expanding the company’s enterprise-y services. With this much investment, some sort of corporate transaction is sure pending in the future — as an acquisition, or more likely, an IPO.

Today though, the company brags about how few people (250+) it employs relative to its footprint. To be sure, Automattic has run admirably lean while competitors sink major resources into pitiful marketing campaigns. As a customer, however, you need to understand that you will be dealing with a smallish company with major growth (read: turbulent) years that lie ahead.

Over the years, Automattic has made a few modest acquisitions, such as IntenseDebate, which provides threaded commenting, and PollDaddy, a popular poll service. These make sense from a functional standpoint. They're popular with many WordPress users, and if anything, they emphasize Automattic is still firmly focused on blogging as its core business.

Strategy & Roadmap

In development, new functionality still tends to take precedence over fixing bugs. This means frequent releases, where major functional updates are quickly succeeded by bug fixes. In practice, this means that you should be prepared to upgrade regularly. Don't expect to be able to plan this ahead, as updates are rarely scheduled. While the software is very easy to install and start using, you should get expert admin assistance to help you understand how to maximize your WordPress implementation.

Like most vendors, Automattic has been a bit inconsistent with its roadmap. Today, you can find a roadmap for a new version 4.0 (and 4.1), but not all the contemplated features and changes are guaranteed to get included in the final builds. Expect the new version in Q4 2014, with the first dot-release slated for 2015. As always, you're likely to see interim security patches.

Automattic says version 4 will include the following changes:

- In-context editing
- A more stable JSON API to publish content to native mobile apps (among other things)
- Better asset library browsing (which is helpful for larger sites)
- Simpler and more transparent language pack applications

On the whole, version 4 will be more of a continuation of the 3.x trunk — saving Automattic from doing a 3.10 — rather than a major overhaul of the platform. Customers seeking more enterprise-y features (like a better API into the core platform) may be disappointed, but those valuing continuity more than complexity and richness will likely be pleased.

Support & Community

In general, the company, developers, and product are highly regarded. WordPress boasts an unusually active and vibrant community, with a variety of key players whose motives range from altruistic to pecuniary. As a practical matter, it means you may need a guide — not unlike navigating SharePoint or Drupal ecosystems — to help separate the good advice from the bad, and sift through the cornucopia of themes, plugins, and hosting environments.

Licensing

Licensing models and costs are listed in the chart of the Introduction to this chapter. WordPress is available as open source (GNU public license), which you can install in your datacenter or almost any major cloud environment. Automattic offers a spectrum of hosting and support options (as do other firms — large and small — from around the globe).

Note that more advanced Automattic services are quite expensive. The company seems to want to capture the very lowest and highest ends of the market (the latter probably suited best to media companies). At the high end, you pay dearly for extra developers or faster (less than a day) support times.

What if you are in-between — not too large, but not small? Given that WordPress information is easy to come by (and for a seasoned developer, the source is easy to understand), we recommend that you look for experienced local support.

Conclusion

WordPress is a product universally liked by its users, and offers many features out of the box. If you want something that isn't in the basic install, you can modify the open source code; but in all likelihood (as with most other open source platforms), someone has already written a plugin to accomplish what you need.

First-time setup is a breeze, although upgrades are frequent (especially owing to security problems), and a lot more difficult. If you have some knowledge of PHP, then designing your own templates for the software isn't too hard. However, remember that WordPress is firmly rooted in blogging (its core business), and has only recently been used for more general-purpose web content management. This means that it lacks finesse; you'll find no easy way to organize pages in a site tree, and there is no substantial access control, workflow, or globalization. In its basic form, there are only two — very similar — content types; adding different content types was never really part of the roadmap.

The advantage WordPress has over Drupal is that users prefer the WordPress interface. The advantage over DotNetNuke is that WordPress has a much larger and active community. Like other popular open source systems, however, you'll need to stay on top of (frequent) security patches. Other enterprise-y features like workflow and user entitlements are also lacking.

You could do well with WordPress as a CMS if you keep its limitations in mind, and avoid complex scenarios. Ultimately, the product remains a best-of-breed tool for blogging, but it can moonlight somewhat for managing web experiences — particularly as a dedicated microsite platform for enterprise customers who go in with their eyes wide open to the product's manifold shortcomings.

If you have hands-on experience with this product and wish to share your feedback, please write to us at feedback@realstorygroup.com. All customer input is kept confidential.

Do It Yourself?

In the face of confusion over CMS package choices and what appear to be steep licensing costs and common stories of failure, some enterprises elect to build their own. Frustration with existing implementations — especially among IT groups — has also created a demand for do-it-yourself development. The case for do-it-yourself (“DIY”) — or “bespoke” — solutions is that theoretically they can be completely customized to your exact needs.

Any perceived advantages carry the seeds of their own problems. A system built specifically for your company can get out of control — both from a budget and scheduling standpoint — if developing applications is not your core competency. Homegrown systems often strain under their own weight of cobbled-together features. Whatever competitive advantages it may provide at launch, a DIY platform typically can’t keep up with the rapid pace of innovation found in packaged solutions.

From a total cost-of-ownership perspective, the financial benefits of DIY may be illusory, as well. Companies frequently overlook the fact that in-house systems require committing valuable resources and incur their own costs in maintenance and enhancement. Sure, package vendors may be insecure, but how long is your lead developer likely to remain with you? If the internal team turns over, who else can you lean on for help? Hopefully you employed commonly known technologies, but there is no DIY user group, and if there were, it would be a Tower of Babel.

Another common failing of DIY content management applications is that they typically don’t provide a highly usable visual interface for non-technical administrators and end users. Developers argue that hiding complexity from end users — if executed poorly — can lead to inflexible applications, and they are right. However, given the choice, end users care a lot about expediency and really prefer to use simple GUIs; they care more about the content itself and less about the mechanics of managing it. Even low-end commercial products represent the fruits of thousands of person-hours in usability testing, feature enhancements, and debugging. *DIY systems tend to be friendlier for webmasters than line-of-business contributors.*

To be sure, there is a case for DIY efforts in prototyping possible approaches; testing concepts and systems change on particular sets of users. We don’t advise going into production on critical systems with something homegrown, especially now that lower-end CMS products have become so capable, while higher-end CMS products have become more extensible. The only possible exception is within high-end, high-traffic media companies, where very specialized needs have dictated custom solutions.

If you absolutely can’t stomach licensing costs but want a supported system that can evolve and hold up in a production environment, then consider at least starting out with an established open source platform (see the Open Source Vendors section).

In the end, however, you may have found it less costly to implement a low-cost commercial CMS.

Regional and Niche Players

If you are looking at mid-market or lower solutions, you also might wish to consider what Real Story Group calls “Regional and Niche Players” as part of your list.

The problem is finding and vetting them. Once you get beyond the top 36 vendors from this report that have an international or at least continental footprint (e.g., North American, pan-European, or Asia-Pac), a prospective buyer must wade into a sea of perhaps 300–400 second-tier vendors with a largely regional (in North America) or national (Europe and Asia-Pac) presence. These are “Regional and Niche Players.”

A healthy number of regional and niche Players have strong offerings, but they tend to be severely constrained in their ability to sell to — let alone support — customers outside of their region, or a specific vertical niche. In exchange for the intimacy and other benefits of working with a smaller provider, you also run the risk of the firm diverting its attention elsewhere, rather than committing wholly behind its CMS offering. Still, regional and niche players tend to fit a profile that makes them suitable for certain buyers in their locale.

Regional and niche players:

- Tend to be newer providers, having evolved from services firms or agencies, and usually number 10–50 employees, typically with only 6–10 working on the CMS product, itself
- Typically survive with a majority of revenues (60–75 percent) from services work, despite ongoing attempts to morph into software companies
- Usually build out other modules (e.g., polling, surveys, and newsletters) around the CMS product to own more of the customer base and avoid complex integration work
- Rely on one or two big clients to help build the product roadmap, and then re-market it to numerous smaller prospects
- Can boast a smattering of clients from outside their regional base, giving them ambitions of going national or international
- Have largely unfulfilled expansion plans via integrator and reseller channels to compensate for a lack of sales acumen, resources, and geographic breadth
- Often offer hosted/SaaS solutions to cover smaller clients
- Appear to be nominally profitable (or at least cash-flow positive), largely due to the success of their professional services work

So how do you find some good regional and niche players to add to your shortlist of bidders? Talk to peers in your region, and find out what vendors they use, or solutions that were proposed to them that looked attractive.

Some of the regional and niche players worth reviewing include:

- **BrightSpot** – A small, Washington DC-based vendor with a growing following among media and entertainment companies. The product is unlikely to fit non-media scenarios for companies outside of North America.
- **Bridgeline Digital** – Stemming from its services/agencyware roots, the vendor has been making halting attempts to become a traditional software vendor in North America

- **Contensis** – Developed by Zengenti, a UK-based vendor, which has a sizeable focus on professional services for web development, as well as its CMS product that has been implemented by some government and educational institutions
- **Elcom** – Australian vendor with focus on Intranets/Digital Workplace, but also venturing into the WCM space
- **ExpressionEngine** – A lightweight PHP product with a small but loyal following
- **GX Software** – Dutch makers of XperienCentral CMS have a loyal following in the Netherlands and some other parts of Europe
- **Gentics** – Viennese vendor with presence primarily in the DACH region, most known for its rich text editor Aloha (which commercial vendors like Ektron OEM)
- **Goss** – UK CMS player boasting a large number of clients in the public sector.
- **Jahia** – French company and developers of a Java CMS + portal offering. Jahia has been trying to expand into the US market for a few years without much success, possibly due to its meandering roadmap.
- **Movable Type** – Venerable blogging tool known for its unusual “baking” (decoupled) architectural model is making a comeback to compete with WordPress
- **Squarespace** – another blog+ platform designed for highly designed (read: agency-driven) sites and microsities
- **Squiz** – Australian consultancy that went the CMS product route, but with little global presence beyond its homeland
- **Veeva** – An example of a highly vertical approach to content management specifically designed for narrow pharma and life sciences scenarios.

As with all vendors, regional and niche players have strong hopes for expansion. However, growth beyond their regional markets may be difficult, particularly in Europe and Asia-Pacific, where it is extremely challenging for a software vendor without a local presence to expand into each national market, even if they are just trying to establish, nurture, and support reseller relationships.

Nevertheless, we believe that this class of vendors as a whole is quietly doing quite well right now, partly because of increased demand for CMS software products that cost US \$20–60,000. We believe that this segment of the marketplace will continue to be active, giving regional and niche players fairly good near-term and mid-term staying power, as well as strong profits and cash flow for those that execute well and are able to control costs. Regional and niche players will remain under threat from national and international CMS vendors like Ektron, Sitecore, and Microsoft. These vendors have also targeted the mid-market, but have not executed comprehensively on the sales side — yet.

In the meantime, you should perform as careful — if not more careful — due diligence on this class of vendors, just as you would a CMS-focused solutions provider.

Part 5 – Advice, Pitfalls, and Best Practices

How to Select a Web CMS Package

There is no special magic to selecting the right Web CMS package. You should take the same approach as you would in selecting any major server software. One difference, though, may be the breadth of internal users and external visitors who will interact with your CMS. The more you can efficiently involve those disparate players, the greater your chances of success.

In fact, the steps below essentially outline how to remake your entire CMS. New tools can play an important role in your efforts, but in order to succeed, technology must serve the business master.

Embrace Stakeholders

Inventory the range of stakeholders for your CMS project. Then identify “internal champions,” people with line responsibilities who can advocate for rebuilding your CMS. Internal champions should not necessarily be the most technical representatives, just the most in pain under the present system or the most to gain from the new one. Typically, those users with the most at stake are editors, authors, product/service managers, and other communicators. Build an internal team to work the project from start to...well, it will never really finish.

Don't forget about your IT group. We sometimes witness selection teams trying to work around their internal systems resources because they don't like the direction the latter is heading. Circumventing your IT group is a grave mistake. They are going to have to support and enhance whichever package you choose, and none of these products is completely “out of the box.” In any enterprise software selection process, if your Marketing, IT, and other teams have strong disagreements, it's always best to surface and address the conflicts as early as possible. Note also that you may need to involve different IT groups, if your company splits responsibility for application development, network and server administration, and security. They are crucial to a content management project.

Likewise, CMS technology choices and system design should never be left solely to technical stakeholders. Sometimes marketing groups abdicate responsibility for solution selection to IT groups because the former does not have sufficient time or expertise. This is a mistake: The wrong tool will eat up much more time and money than a committed presence in the selection process.

One of the other key players is your executive steering committee — or its equivalent. Involve them early and often, and make sure to get their continual executive support. Among other benefits, they can help you build a business case and advocate for a reasonable budget.

Build and Prioritize Your Initial Requirements

Use the questions and checklists in the previous sections of this report. Take a thorough inventory of content areas and owners (you'll need to do this eventually), and base-level corporate infrastructure (human, knowledge, physical) for the project. A complete enterprise content audit may be overkill — focus instead on the major content components going to your

key web properties. However, look deeply at those components, because what you find may surprise you.

The natural inclination is to develop expansive requirements. This is OK as a starting point. At each step in the process, you'll want to ask your team, "How important is it really that this aspect be automated?" Remember you are not building the perfect system, just a great one. Furthermore, as you saw in the product reviews, no single platform presently on the market can address the entire Web content lifecycle seamlessly. Remember that complete requirements are nice, but prioritized requirements are gold. Knowing which features or attributes are more important than others will make it easier to contrast the vendors, and will also help you keep costs in line with your budget.

At this point, you may wish to sound out either your favored integrator or a handful of vendors who would appear to be a suitable match, especially on broad-brush cost estimates, to help inform the next step. This can be done by issuing a "Request for Information" (RFI). Nonetheless, avoid a vendor dog-and-pony show at this point. It is premature and can waste your time (and theirs).

Develop Use Cases or Scenarios

Sometimes called "scenarios," use cases can provide a much richer way of describing your needs and connecting them to business benefits. Use cases can also vastly simplify and speed up the vendor selection process by giving everyone a target to shoot for. Finally, selection team members can understand vendor offerings much better if they discussed and demonstrated using scenarios that team members face every day.

The more detailed your use cases are, the more you will be able to differentiate solutions. However, if time is short, even simple use cases may suffice. Consider this one:

Scenario: Management of time-sensitive content

Scenario: Management of Time-Sensitive Content

User Targeted: Internal Use

Subject Personas: Rosita, a content editor in the PR department
Ricardo, content approver

Scenario Description Ability to author content and schedule it to be published/expired on certain dates, facilitated by a workflow process.

Background: Web CMS internal users must be able to efficiently and quickly create content — without involving IT members. Some content is time sensitive and must be published and retired on certain dates/times. Workflow is an important element to this editorial flow.

Objective: To understand how content is authored, scheduled, and how it goes through a workflow.

Narrative:

Rosita is a PR manager. Her task is to promote the upcoming fifth Annual Tasmania Surfers Competition. She will be sending out a series of press releases related to the event, and would like to create a press kit that goes along with the release, in addition to publishing event information to pages on company.com and microsites that have information about Tasmania.

Rosita is a frequent computer user; she uses email, Word, and PowerPoint regularly. She's heard of HTML, but isn't quite sure what it is. She writes all of the content for her web pages offline in Word, applies some simple formatting to them, pastes the content from Word into the CMS, and then reviews and corrects the formatting as necessary. Upon pasting, the system strips out all Microsoft tags from the content, making it clean and proper HTML.

Rosita creates a page in the CMS for the event by entering the details (location, date, and registration info). She then adds a more detailed description of the event outlining a few of this year's highlights, and adds a few pictures from last year's event to make the page more engaging.

She looks over her work in preview, and then starts the workflow that will send it to her manager for approval. Ricardo, her manager, gets email notification that there's content requiring approval. Ricardo logs into the CMS, previews the page, and approves it. The page is automatically published to the event calendar, or sent back with comments so Rosita can make needed adjustments. Rosita also sets an expiration date on the event so that it will be removed from the website the day after the event.

A week later, Rosita creates a press kit page, embedding one new video and reusing two older ones, with links to two case studies already on the site. She adds bios and headshots for two of the featured surfers at the event. Since the copy for all of these items was already approved by her manager Ricardo via email or via CMS, she schedules the press release and kit to publish at 7am ET 48 hours later. She sets the expiration date for the press kit to be a week after the event, in case anyone writing about the event afterwards needs the information.

Rosita wants to feature her event in the newsroom section of the site a week before it starts, so she sends a request to Corporate PR of the email to be featured. A Corporate PR manager programs a short description, links to the event, and the press kit, and the newest video to the newsroom-featured content area, and sends the page to Rosita through the CMS workflow for her review. Since Rosita has editor permissions, she makes her final changes to the page, and sends it back to the Corporate PR manager so that he can schedule it for publishing. The Corporate PR manager schedules this feature to expire on the last day of the event, which then will be replaced by default information.

Whatever you do, avoid “check box” RFPs. Vendors have seen them all and have figured out how to check all the boxes. Yes, you still need to outline some canonical requirements (e.g., “must run on Windows server”), but try to keep these to a minimum and instead describe more use cases to illuminate the breadth of your needs. Remember the differentiating power of

“how.” Rather than ask a vendor, “Do you support WYSIWYG editing?” Ask: “How do you support WYSIWYG editing?”

There’s one final point when you change your RFP to be less about the product and more about your firm and its needs. To the extent that you open up your business processes, failures and successes in great detail — and ideally you will — it is prudent to have potential bidders sign a strict non-disclosure agreement first.

Solidify Your Business Case

Before you move on, make sure you can still connect a new CMS to your broader corporate objectives using one or some combinations of the rationales outlined in the first section of this report. Having even a simple case statement that does this will facilitate all the choices you make going forward.

Is this a good time to set a project budget? Two schools of thought predominate here. One school recommends waiting to see what potential opportunities lay ahead and setting a budget later. However, it may be more practical to set a budget now to discipline the rest of the requirements gathering and vendor winnowing, while remaining flexible depending on what arises down the road. In either event, don’t forget about customization and integration resources. Account for all of the services you may need, and only set aside 25-35 percent of your budget for software.

Design the Outlines of the System

You can accomplish several tasks here to scope out the likely size of the project. Hopefully, the use case building exercise has compelled you to model your existing workflows and review what aspects can be better automated. Now take a stab at building taxonomies and vocabularies. Diagram how you anticipate different systems will work together. Create the information architecture for your new site, along with wireframed page component diagrams. Develop a creative brief as necessary, and perhaps user interface prototypes.

Many well-intentioned project teams skip or downplay this step. You should not. A deep understanding of your own content is an essential precondition to managing it more effectively.

You may need some outside help here. The key in selecting external consultants is their solution neutrality. An integrator may well define the solution in a way that makes it most susceptible to resolution through their favored technology, and may receive a bounty from the CMS software vendor for recommending their product.

Review Technology Alternatives

Use your requirements and the outlines of a new system to develop a workplan, which will simplify matters for you and vendor alike. The workplan, together with your use cases and the deliverables described above, can serve as the basis for an RFP.

But which vendors should you invite to that process? Well, this report answers that question, and provides comparative charts with narrative explanations. If you are an RSG subscriber, also be sure to explore our [Custom ShortList Builder](#). Vendors will gladly come in to present their products first-hand, which may serve as a learning experience and could help sharpen your requirements and shorten your list. However, chances are that if you have designed your system carefully and digested this report, you might not need to conduct a wide cattle call.

Have them demo your use cases rather than their canned routines. A browser with a good Internet connection is much more useful here than PowerPoint slides. Insist on the former.

Be sure to ask what optional modules the vendor would *not* include in your solution and is *not* including in its price. Unfortunately, “sandbagging” remains rampant: After you are underway, the vendor informs you that you must purchase optional modules. This potential for extra costs is especially important to consider for “deployment” modules and application server or portal connectors.

The bottom line here is that you should not be seriously comparing vendors until you have completed all of the previous stages.

Perform Due Diligence

You probably already know the importance of performing due diligence on server software, but in any case, here’s a brief primer. For starters, you’ll want to talk to companies like yours who have implemented the package. Vendors will total the number of licenses they have issued in the past year, but try to find out how many actually implemented and then kept the product. If you are suspect about a particular feature or module, ask to see it in production on a live server from among the vendor’s clients. This is especially critical when your implementation depends on integrating two or more products — the fact that the vendors are “partners” means little. How many sites currently use those two products in tandem now? Fair or not, the early adopters provide integration lessons for everyone else.

If you are going to procure professional services from the vendor or an integrator (and you probably will), be sure to meet the team who would work with you. Every company has experienced CMS hands. Meet the actual project managers, architects, and developers who will be implementing your system or training you how to run it. If you can’t work successfully with them, what makes you think you can work with their product? Note that this is especially important with vendors who are enjoying substantial success and momentum in the marketplace, and are therefore growing (and hiring) a lot. That’s right: financial health and the quality of human resources at hand can be competing dimensions.

Review any vendor’s financial performance, but with grain of salt. We have seen big, successful companies jettison CMS offerings, and niche vendors survive and thrive. Nevertheless, there remain key metrics to evaluate, like available cash (and burn-rates for cash-flow-negative companies) and revenue momentum. Of the two, cash on hand is the best predictor of whether a company will survive. Use your experience — and instincts — here. For better or worse, positive “buzz” around a particular vendor may be a better predictor of long-term viability than the quality of the technology.

Download Trial Packages When Possible

You can sometimes download trial versions of CMS packages from vendor websites. Note that many of the trial versions are “light” implementations that can be buggy and not always well supported. However, they will give you a sense for the product. Of course, with an open source package, you can always download, install, and play with the complete production version. Use trial versions *as a basis for asking questions, rather than as your final decision point.*

Remember that you may need more than one package as part of a complete solution. Your requirements may push you to consider XML-handling tools, search engines, a separate

delivery platform, automated classifications products, and perhaps other software, in conjunction with a CMS package.

Evaluate Security Implications

By now, your security staff should have a good understanding of how the different products will fit into your network environment and overall protection schemes.

However, the CMS package itself should be scrutinized. Be sure your vendor can answer these six key security questions¹:

- Who runs security tests on the CMS? “No one” is a bad answer.
- Are there regular security notifications? When breaches are found, you’ll want to be the first to know.
- Which third-party products are included within the CMS? Search engines and other applications have their own security considerations.
- Are there security specific guidelines for the CMS? If not, they probably haven’t looked deeply enough.
- Can the supplier provide staff with recognized security qualifications?
- Have you established adequate internal controls? Remember that most security breaches are internal.

Require a Proof of Concept

If you are about to drop \$US 50-500,000 on CMS licenses alone (with integration on top), it’s reasonable to ask your list of vendor finalists to come in to your shop, install their packages on your hardware, and develop some sample templates and workflows that make sense to your team. You should define this exercise in advance, using your scenarios, rather than using canned vendor demo sites. Note that, depending on whether the proofs can be done concurrently, a proof-of-concept step could add a month or more to your selection cycle. Nevertheless, this kind of “show-me” evidence can be a powerful predictor of which offering will work best for you, and in our experience, it saves time in the end.

Perhaps most importantly, it is the best way we know for actual system users to definitively identify what will work best for them. Don’t be scared by this. It is to your benefit to emplace accountability with contributors.

You may risk ruling out a vendor who elects not to compete in this way. For an enterprise installation, this is an unreasonable position for a vendor to take, but in the mid-market, the cost of participating in a proof of concept could far outweigh any margin the vendor would like to achieve. If your budget totals \$US 50,000, you may get the hungriest vendors to send three developers in-house with you for a week, but perhaps not the *best* vendors. Software companies that repeatedly lose money will soon go out of business, especially if they are privately held.

One way to avoid this is to offer to compensate the vendor an agreed-upon sum for their time (but not their software) if they invest in a proof of concept and do not emerge the winner.

1. We are indebted to Colin Cornelius (colin@guinet.com) for this analysis.

If your budget runs into six figures (with attendant complexity), you *definitely* want to do a competitive POC (sometimes called a “bake-off” or “sandbox”). To be sure, managing this requires some attention and resources on your end. Here’s a simple checklist of things to consider, assuming you have two finalists:

- Two sets of development space for their developers, ideally each with their own conference room; they should bring their own laptops
- Ideally, one developer from your team to work with each vendor or at least check in with them at the end of the day, to see how they did, and *what* they did
- A project manager on your end to communicate with both teams (divvying time fairly), with availability for Q&A
- Wireframes and other supporting documentation as possible to support your scenarios
- Your selection team on hand to test the POCs at key points, including an important mid-project review
- Two sets of infrastructure (they will need different configuration settings and won’t be able to share)
- Programmatic access to other essential internal systems, such as your LDAP repository, and your existing web content, designs, and code as necessary
- Access to stress- and load-testing tools
- A strict deadline for both parties (usually 2–4 weeks, but more involved projects may need more — some vendors will finish in less)

If you can’t undertake a proof of concept, then at least send your staff — developers *and* authors — to training at the two vendor finalists’ training centers. You’ll likely have to pay for it, but it will be a fast way to feel certain about which solution will best suit your needs.

Choose a Package

With solid requirements and a firm budget the choice should be clear by now. Negotiate your best terms. (See “Vendor Pricing” on page 843.) If you go the proof-of-concept route, you probably want to negotiate those terms *before* the final tests, when you will still have maximum leverage, but hold off on completing the contract, as issues may arrive during that phase that you might wish to address in the final agreement.

No software is perfect. Conduct a group debrief after the competition phase and make a list of what you don’t like about the package you are going to select. This helps you set internal expectations and account for necessary workarounds as you finalize the implementation plan. Contributing resources towards resolving some of shortcomings your team finds could be part of the final deal.

Prototype as Early as Possible

If you are purchasing a large system, consider confirming that you made the right choice before getting the full spate of licenses you will ultimately require. This means quickly implementing even a partial solution for part of your online platform — perhaps a single section of your site, or within a single department. That way, you — and your vendor(s) — can learn critical lessons while the stakes remain reasonably low. It also allows you to show project momentum. If it all blows up, hopefully you have hedged some of your investment.

That sounds awful, but what is worse than admitting a mistake and switching to a different package? Not admitting your mistake and being stuck justifying a system that doesn't meet your basic business needs. On several occasions, Real Story Group has heard, "Yes, this software stinks, but we paid \$X hundred thousand for it, so we have to use it..."

Recognize When You Need Help and Seek Outside Assistance

By this point, you may wish you had gotten help. Perhaps the project has stalled or you have not been able to reconcile differences among the team. The right consultant can add value from the earliest stages of the project.

A knowledgeable and experienced outsider can help by:

- Identifying key internal stakeholders
- Facilitating among disparate factions
- Applying proven requirements templates
- Helping to clarify the confusing array of technical and design choices
- Assisting in developing the RFP and evaluating vendors
- Generally pushing the process along (this doesn't have to be a long process)

The right company can also serve as a resource (or outsource) for implementation by a team. As always, picking the right consultant will present a key challenge. You'll be able to choose from among independent contractors to big consulting houses, and everything in between, including public web agencies, regional integrators, and niche design and information architecture firms.

In this market, afford yourself the luxury of picking an individual or firm that clearly understands your company's business objectives and CMS best practices, as well as the CMS marketplace at large.

Note that most professional services firms have preexisting relationships with specific CMS vendors. This is not necessarily a bad thing — for example, you want an integrator to have preexisting experience with any particular package they implement for you — but be sure to discover and understand potential conflicts before getting started.

In any case, be sure the specialists working on your system have worked with the relevant CMS software package(s) at least three times before, and with the current version at least once. Deep knowledge of complex software is an acquired talent. The difference between an experienced and a novice developer can make or break your project schedule — and budget.

If you are looking for an independent outside advisor during your selection process, be sure to [contact us](#) to explore how we can help you.

Vendor Pricing

CMS software prices — like the vendors themselves — are all over the map. In any case, you should remain diligent in your negotiations and plan for expansion carefully; there is no free lunch. Remember that in addition to software, you will almost surely be buying some level of professional services and ongoing support. Let's look at both of these categories.

Software

With some exceptions, the predominant factor in pricing is the number of CMS servers you need to employ, often the total number of CPUs involved, and potentially the heft of those processors. In this connection, don't forget failover and development servers, but remember that you can frequently negotiate for free or substantially reduced licenses here. Note again, however, the distinction between failover servers and load-balanced servers. The latter are running concurrently, and typically require full licensing of all servers, while a failover server that does not kick in until the production server fails should not encumber you with a full extra license. Of course, the vendor may try to charge you for it. Just say "no" in the contract.

In server licensing, the distinction between content Production and Delivery becomes critical. If you physically separate content contribution by internal users from content consumption by external visitors into two separate environments, a tightly coupled CMS will charge you for server licenses on both sets of machines. A decoupled system (like Percussion, HP, et al.) may still charge you for "receiver" licenses on the delivery boxes, but these tend to comprise a much lower expense. Make sure to find out if any receiver agent is priced by server or CPU.

Always keep a sharp eye out for optional modules and their pricing. A common tactic in server software is to demonstrate the advanced version of a product and laud its features in a written proposal, but then price out a baseline version that offers only a scant few of the bells and whistles that enamored business users to the product. In any competitive solicitation, insist that the technical proposal only discuss features and options that are priced in the core cost proposal, and that optional features and modules are clearly identified in both sections. Unfortunately, you can expect vendors to try to muddy the waters here, anyway. Keep going back to them to break it all down. This may be an iterative process, but it is well worth your effort.

In addition to server license costs, some CMS vendors continue to charge by user as well. You'll want to understand if this means concurrent user, rather than named user. Named users represent the total set of individuals with rights to manage content in some way. Some of your named users may only access the system once a month or less. Concurrent users represent the total number of content managers accessing the system simultaneously at any given time. Concurrent usage limits have become increasingly popular, but a named user approach may be more attractive to you if your publishing system encompasses just a few very active contributors anyway, because the per-user fees tend to be lower with this format. Some vendors have tried to finesse the issue by charging separate tiers of user fees for occasional users versus power contributors. We find this unnecessarily confusing, but you may not have a lot of room to negotiate here.

Many CMS vendors boast that they do not charge user fees, but remember; there is no free lunch. These may be some of the same vendors whose products' CPU capacity tends to be lower than their competitors. Here's how this works. If system performance for Vendor "X" tops out at 100 concurrent contributors per CPU (assuming suitable memory), then you have effectively purchased a 100-person user license with each CPU's worth of Vendor X server software you buy. Nevertheless, generally speaking, the absence of user fees tends to make for simpler and more scalable licensing formulas. You'll want to plan your own growth scenarios carefully and decide accordingly.

In planning your growth scenarios, consider how a particular product's scalability matches up against your overall architecture. This is good practice for performance reasons, but will also

affect potential license fee growth as well. For example, if the CMS vendor employs a search engine index as its repository, then likely the size of the index will determine when to get a new server before you run out of processing power for scores of concurrent contributors (because the incremental re-indexing when content is changed or added can become a prohibitive drain on the server). Fortunately, most vendors can provide reasonable scalability metrics, though you should review their assumptions carefully. Some products are more memory intensive, while others consume disproportionate CPU cycles and still others present more of an input-output (“I/O”) strain on a server. As your system expands, you may require a separate machine just for deployment as you push out large bodies of content, or as you experience network bottlenecks. All of these issues carry license implications to the extent that you need to deploy more servers to run the CMS package.

As you might expect, hosted and cloud providers present a more services-oriented fee structure. It is often the case with cloud and SaaS vendors that you are quoted a price that depends on your traffic levels and/or amount of content and storage. Some hosted CMS vendors may also charge according to number of named users or amount of updates per month. This is another area for potential scalability issues as you grow; while the initial quote may appear enticing, plan carefully for the future.

In nearly every case, you can expect to pay for developer kits (SDKs), API kits, or developer seats, or all of the above, and if you have a large development team, the total can become quite a significant financial investment. Fees for developer clients often run around \$US 5,000 per seat.

Professional Services and Support

CMS software vendors will exhibit less flexibility about pricing for their professional services. This is partly because vendors incur real incremental labor costs here, and partly because they have become addicted to the predictable profit margins from their services work amid an otherwise competitive market for their software. Most CMS vendors now offer common service “packages” — such as “installation,” “quick-start,” “content analysis,” and so forth — that are based on specific performance measures rather than explicit hourly rates. This is good business practice, as it enables them to leverage repeatable processes they’ve learned over the years and price them according to perceived value (at what represents in fact very high hourly rates). We see nothing wrong with this if you are paying for true expertise, but you should insist on the provision of experienced practitioners in any final contractual arrangement. Moreover, wherever there is value-based pricing, you have room to negotiate a better deal.

You should budget for support and maintenance costs ranging from 12–24 percent of the price of all software you purchased, on an annual basis. Make sure this covers upgrades, including major upgrades, as some vendors have tried to argue that particular major version upgrades represent “new” products. Don’t put up with that. (At the same time, don’t expect that covered software upgrades mean no cost to you, since larger installations typically require professional services help from the vendor to migrate to the new version — after all, who else knows how?)

One factor in varying maintenance costs is that most vendors now offer separate tiers of support at different price levels. Buyers tend to underestimate their support needs; remember that application servers can go down at any time. One strategy we recommend is to seek a higher tier of service, but negotiate the rate down. After a contract is signed, the vendor will quickly want to deliver the software. This is partly so that they can invoice you, but also, many contracts state that the support meter starts ticking the day you take delivery, unless you pre-

negotiate a later install date. If you don't install the software until three months after you bought it, you may have paid a year's worth of support for only nine months' worth of coverage.

During the product selection process, find out how the competing bidders recognize or “book” revenue internally. Some publicly traded vendors don't put revenue on their books until the customer's implementation goes live. We think this is a good thing for buyers, sellers, and investors alike. However, such a vendor may pressure you to do quick “pilot” or “reference” implementation before you are ready. See whether their reporting incentives synchronize with your schedule.

Expect to incur other expenses as well — some negligible, some large. Training can represent a sizable expense. Developers definitely require instruction from the vendor, and they may need to travel to get it. Some CMS vendors are offering various e-learning opportunities, such as how-to videos, webinars, and online courses. We think this is a welcome development for both parties, but we suggest you send developers to in-person classes first. Let your technical team rub (and bend) elbows with vendor techies — that's often when the best information is transferred.

The instruction of end users of the CMS is a different matter. The vendor is typically not the best party to conduct training for content contributors and editors, unless they are the ones most familiar with your implementation and can successfully customize their default curriculum accordingly. Remember that you are not instructing business users on a particular software product, rather you are helping them to use your CMS that happens to use a particular product underneath the covers. The latter is something different altogether.

Some Final Words on Pricing

Like everything else, buying at the end of a quota cycle (typically this is quarterly) can lead to better deals. However, don't let the crush to consummate a quick contract in exchange for a discount push you into committing prematurely — we see just as many (or more) buyers lose out in hasty deals as we see for vendors. Understand fully *what* you are buying, what you are *not* buying, and on what specific terms.

Twelve Common Pitfalls to Avoid (and Best Practices to Follow)

Throughout this process, you can take advantage of the lessons learned by those who went before you. Having led several CMS implementations and after debriefing dozens of practitioners, Real Story Group has seen much go wrong — and much go right. Here is a distillation of common pitfalls, along with best practices to follow to improve your likelihood of success.

Pitfall 1	Best Practice
Selecting a Web CMS package <i>before</i> developing solid requirements and a business case	Gather thorough requirements — but only after establishing a business case to shape and discipline the process — <i>then</i> select a package that truly meets your needs.

Talk to ten software integrators and nine of them will tell you that the biggest single predictor of a failed implementation is when a company chooses a specific package before requirements have been adequately established. The second biggest predictor? A requirements phase that drags on because no unified business strategy exists against which tough choices can be tested and measured.

When companies select CMS packages before they agree on business objectives and adequately plumb stakeholder needs, the technology inevitably ends up driving the system — the people, the business rules, the editorial processes, even the content itself — rather than the other way around. You’ve probably worked amid systems like that, and you probably didn’t like it.

Perhaps more importantly, you will find along the way that you have critical forks in the road with important decisions to make. Do we impel authors to use forms-based entry or focus on Word conversion? Do we integrate our catalog content using special connectors, or not? There are no simple answers to such questions, but a clear business rationale will help you make solid choices.

Pitfall 2	Best Practice
Not getting a clear mandate from the top	Obtain strategic direction, a suitable budget, and a mandate for necessary changes.

Like anything in business that yields a strategic payoff, implementing a new CMS is hard, and potentially expensive. It can bring change to many parts of an organization, and however felicitous the improvements, change unnerves most people most of the time (See Pitfall #11).

Leadership is needed to see a CMS project through. When difficult choices present themselves, being able to refer to the business rationale will lead to faster and more effective decision making. This may require carrots as well as sticks, and you will almost surely need your management to provide both.

Pitfall 3	Best Practice
Thinking a web content management package will provide a CMS	Model your existing and prospective CMS using building blocks of people, content, practices, and infrastructure.

Your CMS software is just one part of the infrastructure building block in your CMS. If your Content Management System is like most others, it will consist of 20 percent technology and 80 percent process.

This is actually good news. You can control and modify a system. You can't always control and modify packaged software as much as you would wish.

Pitfall 4	Best Practice
Not involving internal stakeholders from the very beginning	Involve system users in the design, implementation, and testing of the system.

There is a temptation among IT and Marketing departments to shield domain specialists from key decisions on information architecture and workflow. However well intended, this is a mistake. Only the content owners themselves can put meaning and context to the information that is being published, so they must be engaged in the design of taxonomies, sitemaps, and even page component layouts. If they participate in the selection of the solution, they will be much more likely to participate actively in “selling” it to the rest of the enterprise and engage fully for a successful implementation.

Also, there is a growing consensus that the most usable applications result from a truly user-centered design process, with users involved from the very beginning in the shape of the ultimate solution.

Note that sometimes there are inherent conflicts between capability and usability. For example, the distinction between “power users” and “casual contributors” that is widely held among CMS vendors can become immediately problematic within your particular enterprise. Is a power user an author who needs an efficient interface to accomplish the same thing repeatedly and often, or a kind of managing editor who needs a control panel to accomplish a variety of oversight tasks, such as moving pages and sections or administering taxonomies? Those two very different personas will likely find comfort in very different interfaces. Your high-powered editors may come to despise the “super-user” interface that was ostensibly designed for their needs. Very often simple works better; that's good news for developers, but it places a premium on finding the specific product whose “out-of-the-box” orientation happens to match your needs.

Pitfall 5	Best Practice
------------------	----------------------

Involving *only* internal stakeholders throughout the project

Integrate content end users early and often.

The whole purpose of investing in taxonomies, search engines, and the content itself is to improve the user experience and enable content consumers to realize greater value from your offerings. Test the likely return on your investment by including your best customers and partners in the planning and debugging of the system wherever possible.

The advent of low-cost website usability methodologies calls even more for doing this early and often.

Pitfall 6	Best Practice
------------------	----------------------

Spending insufficient effort describing and organizing content, and underestimating migration times

Invest in mapping the structure of your content, “chunking” it, building workable taxonomies, and creating user-centric information architectures. Then clean your data.

Inadequate work up front in information design and mapping inevitably means leaving key architectural decisions to engineers or graphic designers. How will your news articles be linked to your product line? Technicians and designers have important things to say on that subject, but ultimately, it is a business-information decision.

Also, don’t overlook migration. If you already have a website, you already have web content, and migration of some kind will be in order before re-launching with a new CMS. Most engineers quite properly view migrations of this sort like an extended trip to the dentist. Some CMS packages have migration wizards to import your existing HTML (see for example, the migration tool from OpenText in Figure 444). Others employ Perl scripts to do the same thing.

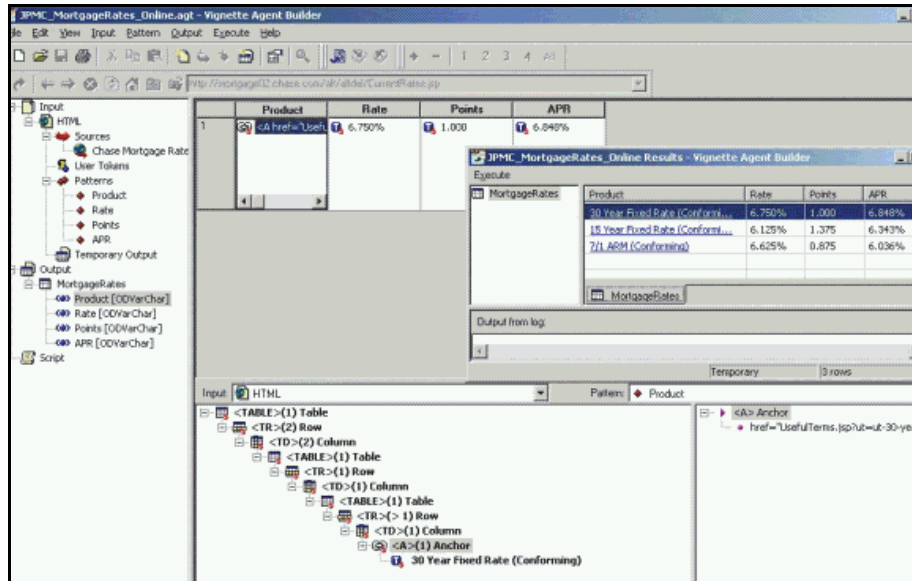


Figure 444. OpenText WEM migration tool.

However, these tools will work well only to the extent your existing content (HTML or not) is structured and *consistently* organized. Migrating databased content would appear somewhat less painful on the surface, but in practice brings its own headaches, particularly if the new system employs a radically different data model. Some good “cleaning” you can start today is to make sure that all of your existing pages are XHTML compliant.

Pitfall 7	Best Practice
Picking a CMS package that doesn't play well with other company applications	Identify the broader IT environment for your CMS effort, and anticipate integration needs.

Since your CMS may evolve into the prime public gateway into your legacy systems (CRM, ERP, product databases, etc.), the ability to integrate successfully is critical. What languages can you or your consultants use to customize the system? How much of the underlying structure and logic contains open APIs?

One way to help yourself is to follow industry standards. Open database models, separation of data and logic into different tiers, and the adherence to industry standards (like CMIS, WEMI, and JCR) can all help protect your investment and increase your rebounding time if the package you implement suddenly becomes discontinued. Remember, though, that standards “adherence” is a relative term, not a state of being. Find out *how* your vendor adheres to standards that are important to you, and map that against the goals you are trying to achieve through standardization.

Also, remember that your IT department will have to support this infrastructure, albeit possibly with outside vendor or integrator help. It should run on operating systems and databases that they can maintain (or reasonably *learn* to maintain).

Pitfall 8	Best Practice
Underestimating hardware needs	Get more server power than you need today.

CMS packages — even those at the departmental level — are notoriously resource intensive. Also, if you are being very creative with things like personalization and metadata, your database will fill up with all kinds of interesting and potentially useful records that could grow to dwarf the storage size of your actual content. You do yourself no good to design and implement the perfect CMS, only to find that your server performance is prohibitively slow for internal and external users alike.

That’s why clustering and load balancing aren’t just for the enterprise CMS installation anymore. If you are a mid-sized company or single business unit, you may still need those features. Make sure your CMS vendor and IT department are on board.

Pitfall 9	Best Practice
Underestimating integration and other professional service needs	Anticipate the need for outside help and budget for it.

The famous last words in any CMS marketing materials are “out of the box.” Even the most prepackaged solutions described in the product comparisons section require some level of integration. For example, Microsoft lauds the ease of customization of its own CMS Server, but notes that it expects that nine out of ten buyers will need add-on services.

Behind some of the stories one hears of failed CMS implementations are companies that spent their entire budget on software licensing and maintenance, leaving nothing remaining for the most crucial part: Actually getting the system to work.

If content is central to who you are as a company, then how you manage, organize, and present it will be different from other companies. Figuring out just what needs to be customized can take almost as much effort as the actual integration. The complexity and dynamics of your business rules and processes here will be key determining factors in the extent of integration required. Also, someone will also need to code workarounds to the inevitable bugs and undocumented product shortcomings (even the best software has both). The key, of course, is to involve enough of your own technical specialists, to ensure that sufficient expertise transfers in-house, so they can make updates and changes to the CMS going forward.

You should also seriously consider who will take care of other tasks that have everything to do with unleashing the value of your content, but only tangentially related to implementing a CMS package — like outlining a meaningful site information architecture, creating value-added visual presentation schemes, and developing user-centered navigation.

Pitfall 10	Best Practice
Looking solely at the product and not enough at the vendor	Perform as much due diligence on the vendor company as you do their product.

When you purchase software, at some level you are “marrying” your CMS software vendor. Companies, like people, have personalities, and software companies often have strong personalities. Understand how your prospective vendor generally behaves, because you are going to be living with them for at least several years. How do they treat their user group(s)? How can smaller customers get attention? How many mergers have they experienced in the past two years (a predictor of internal disarray)?

Moreover, you may be working very closely with their professional services and support staff. Do they mesh well with your team? That chemistry alone could well have a greater impact on the success of your project than the suitability of the product.

Pitfall 11	Best Practice
Missing or underestimating internal change management issues	Recognize and clarify how new systems and tools are likely to affect people’s jobs and enlist people’s support for productive change.

Providing publishing tools to line-of-business managers will be exhilarating and liberating to some, horrifying to others. Some content owners might not *want* to take direct responsibility and control for the timeliness and detail of their part of a website. Workflow tools and detailed auditing mechanisms in particular bring a whole new dimension to accountability. Others may resent finding themselves suddenly unable to dictate the look and feel of their content.

Of course, that’s just the idea. If the Internet is central to the way you do business, and content lies at the foundation of your web efforts, then you want a system of incentives and controls to make sure content is managed well. You are not necessarily eliminating people’s jobs; you’re just asking staff people to focus more intently on their particular areas of expertise. Involve your best people in the design process and the new system should work well for you.

If you have successfully devolved content maintenance to content owners, one of the first people who may need to reinvent themselves is the corporate webmaster. There is, however, a good specialization story to tell. Instead of spending time shoveling content through HTML converters or tracking down the correct version of the latest press release, webmasters can free themselves to focus on higher value-added specialties:

- Content-oriented webmasters can focus on the editorial quality, web suitability, and the substance of the subject matter, itself.
- Technically-oriented webmasters can grow to manage a true online publishing system.
- Design-oriented webmasters can focus on the visual effectiveness of the presentation, and multiply the value of their efforts dramatically by propagating their designs through broadly used (and enforced) templates.

So what goes away here? The generalist webmaster position.

In any event, many people’s jobs are likely to change. Some will have more work; some will have less. Overall, you should be able to manage more content, faster, and with better results, but leadership will be needed to address the inevitable individual winnings and losses along the way.

Pitfall 12	Best Practice
Not preparing for a potential content expansion and change	Anticipate future content expansion in your site architecture.

When people find publishing easier, they will often generate more content. That can be a good thing if you’ve put in place a solid quality-control scheme. Just make sure your systems will scale for it.

One system that can be stressed very quickly is the site information architecture and navigation scheme. New types of documents and heavier volumes of content can tilt the balance of the user experience away from ease of use towards clutter and confusion. Debrief content owners during the prototyping phase to anticipate and address future information flows.

Glossary

AJAX

AJAX is short-hand for “Asynchronous JavaScript + XML.” In practice, it represents a method of updating a web page without doing a complete page refresh. This allows interface developers to create client-server-like behavior in a web browser, with such features as autosuggest form fields and drag-and-drop.

Apache

A popular web server that is freely available under an open source license. The current version runs on most UNIX-based operating systems, as well as on Windows. It is estimated that more than 60 percent of all websites run on Apache servers.

Apache debuted in 1995, and development of the server continues, primarily among a set of volunteer programmers known as the Apache Group. However, the source code can be modified or adapted by anyone.

Application Server

A server program that houses the business logic for an application. Application servers, or “appservers” execute the operations necessary to complete transactions and other interactions between end users and a business’s backend databases and applications.

Application servers provide functionality such as load balancing, database access classes, transaction processing, and messaging. For tiered applications, best practice calls for separating out this application processing from the actual dishing-up of web pages, which is done by a web server operating in front of the appserver.

Application Servers are typically synonymous with J2EE (Java) engines, such as Oracle WebLogic or IBM WebSphere. However, other interactive platforms, such as ColdFusion and PHP, can be seen as lightweight application servers as well.

Application servers, therefore, are key Delivery engines in the CMS space. Some CMS products embed their own application server, while others take advantage of third-party appservers.

Digital Asset Management (DAM)

The purpose of Digital Asset Management (DAM) is to enable companies whose life blood revolves around their media assets — such as media, publishing and entertainment companies — to organize and repurpose those assets to streamline costs and enhance revenues.

DAM systems are especially suited to managing multimedia content, and tend to offer hooks into specialized desktop media authoring systems.

ATOM

An alternative syndication format to RSS (see “RSS” below).

Baking

“Baking” is a colloquial term for pre-generating an HTML page out of components before it is actually deployed to a web server. Typically, this pre-publishing happens at approval time, although it can take place at deployment time as well. Static HTML files can bring numerous benefits (especially around performance), but page assembly is not informed by any runtime user input. Contrast with “frying.”

Binding

In the context of web content management, this is the process of assembling web content by inserting content into an HTML template. This could mean generating a snippet of HTML for an element, or assembling an entire web page by binding multiple elements as part of a holistic layout.

Caching

Caching refers to the process of temporarily storing files in a repository for quick access. When a file, such as a web page, is stored on a cache, it can be accessed without returning to the original server, thus reducing server load and response time. A cache can exist in a section of a computer’s main memory or as an independent high-speed storage device.

Large organizations or institutions use large caches to distribute and update data that is frequently accessed by users. Caching is important for ensuring adequate response times in CMSs because serving custom pages on the fly can be extremely process-intensive, even on low-traffic sites.

ColdFusion

A set of web development products (owned by Adobe) that allows for integration between databases and web pages. Using ColdFusion, a developer can combine a content database with a set of templates to create a site that builds and serves pages “on-the-fly.”

ColdFusion uses its own markup language: CFML.

Controlled Vocabularies

A collection of preferred terms that are used to assist in more precise retrieval of content. Controlled vocabulary terms can be used for populating attribute values during indexing, building labeling systems, and creating style guides and database schema. One type of a controlled vocabulary is a thesaurus.

Decoupled

In a web content management context, it means a complete separation between the content production and content delivery tiers. CMS tools with decoupled architectures are typically delivery-agnostic. That is, they do not provide delivery, or visitor-facing,

capabilities, but instead integrate with web servers or application servers in the delivery tier.

Document Management (DM)

The purpose of Document Management (DM) is to help companies better manage the creation and flow of documents — in particular structured documents — through the help of databases and workflow engines that encapsulate metadata and business rules. Much of what we know about automated editorial workflow comes from the DM world.

A critical drawback to DM products, however, is their limited traditional understanding of content as files, as opposed to discrete chunks of information.

Folksonomy

A folksonomy is a user-generated set of tags or categories; essentially, the social-software trend's answer to the taxonomy. Folksonomic tagging is intended to make a body of content easier to search, discover, and navigate over time. Folksonomy functionality is not inherent to most CMS; folksonomies tend to arise in web-based communities where special provisions are made on the website for users to create and use tags.

Frying

“Frying” is a colloquial term for generating a page dynamically from a repository at runtime when a user clicks on a link. Many CMS packages that deliver content are fundamentally “frying” systems. Contrast with “baking.”

Indexing

The process of creating an index for a database or search engine. A database index associates specific keys or keywords with a unique record. Indexing facilitates the process of locating specific records within a database. Whether and how (and how often) a database, text, or XML repository is indexed can play a significant role in the quality of search results.

Java / J2EE

An object-oriented programming language developed by Sun Microsystems. Java programs are portable across a network, and can be built to run on either the server, browser, or special client.

Java is written in bytecode, a type of source code that must be processed by a Java virtual machine. The virtual machine interprets the bytecode into code that will run within the constraints of a specific computer's hardware system. Therefore, as long as a platform is equipped with a Java virtual machine (as are most of today's servers), Java applications will run on any operating system.

Because of its portability, Java has made substantial inroads into Enterprise computing, and, not surprisingly, commercial WCM vendors have been leveraging it both to expand their own capabilities across multiple platforms as well as integrate with other Java-based systems. The pre-eminent Java standard is “J2EE,” which prescribes a range of functions, but most notably for our purposes, lays out a set of web publishing and

application tiers ranging from Java Server Pages — JSP (for presentation and basic interactivity) to Servlets to Java Beans, where corporate business logic is stored.

JSR 168/286

This is a Java specification for a “portlet,” to allow compliant portlets to run in compliant Portal software applications. In theory, this allows portlets to be usable in different vendors’ portals. In practice, many developers extend portlet functionality such that it can no longer run in a JSR 168/286-compliant container. As a practical matter, more and more Java-based web CMS vendors are making their tools JSR 168/286 compliant so that they can run portlets and become more “portal-like.”

JSR 170/283

Also known as Java Content Repository (or JCR), JSR 170/283 is designed to provide a standardized way of accessing an object-oriented content repository, such that the same set of Java code that accessed the JCR API could work against multiple JSR-compliant repositories. The goal of this standards effort is to separate the repository layer from the middleware. As a practical matter, only a handful of WCM systems are JSR 170/283-compliant. Most have their own unique database access routines.

JavaScript

A relatively simple scripting language that can integrate with HTML code and add interactivity to a web page. It is frequently used for functions that execute in the browser, such as mouse-over effects, calling pop-up windows and alert boxes, and basic mathematical calculations. Developed by Netscape, JavaScript is an open language supported by most current browsers.

While it has some structural and conceptual similarities to the Java programming language, JavaScript was developed independently. Generally speaking, JavaScript lacks Java’s portability and speed.

LDAP (Lightweight Directory Access Protocol)

LDAP is a protocol for accessing online directory services that runs directly over TCP, and is used by nearly all directory clients that are in widespread use.

LDAP directory service is based on a client-server model. One or more LDAP servers contain the data making up the LDAP directory tree. An LDAP client connects to an LDAP server and asks it a question. The server responds with the answer, or with a pointer to where the client can get more information (typically, another LDAP server). No matter which LDAP server a client connects to, it sees the same view of the directory; a name presented to one LDAP server references the same entry it would at another LDAP server. This is an important feature of a global directory service, like LDAP.

Localization

Localization refers to the process of adapting a software product or service for different languages, countries, or cultures.

In addition language considerations, such as support for foreign character sets, localization may require adaptations for currency, time zone, national holidays, cultural assumptions and sensitivities, dialect, color scheme, and general design conventions.

Metadata

A definition or description of data, often described as data about data. For example, the data of a newspaper story is the headline and the story, whereas the metadata describes who wrote it, when and where it was published, and what section of the newspaper it appears in. Metadata can help us determine who content is for and where, how, and when it should appear.

For documents online, important metadata elements include its author, title, date of publication, and subject area.

.NET

.NET is a Microsoft development and application framework that emphasizes Web Services-based integration. It is the successor to Microsoft's COM model.

Object-oriented (OO) database

A database that classifies information as “objects”; that is, structures that include both data and the functions that can be performed on that data. An object-oriented system organizes the classes of objects, the inheritance of class properties, and methods by subclasses and their objects.

Open Source

Open source is a term applied to certain packages and tools, like Linux and Perl, that are distributed free of charge under a license that guarantees the right to read, redistribute, modify, and use the software (source code and all) freely. In terms of content management, there are two broad types of products: community open source and commercial open source. In reality, most open source WCM tools fall on a spectrum between the two poles.

Parbaking

“Parbaking” is a colloquial term for pre-generating parts of an HTML page or website in advance of deploying it from a content development environment to a delivery environment. Contrast with “baking” and “frying.”

Perl (Practical Extraction and Report Language)

A script programming language designed specifically for processing text. Developed by Larry Wall and introduced in 1987, Perl combines syntax from several UNIX utilities and languages. It is often used to write common gateway interface (CGI) programs — one method through which developers can provide dynamic interaction between users and websites.

Perl is a popular choice for programming server-side tasks such as automatically updating user accounts and newsgroup postings, processing removal requests,

synchronizing databases, and generating reports. Its text-processing prowess makes it a frequent choice for building homegrown CMSs.

PHP

An open source script language that was designed specifically to generate dynamic web pages. PHP script is embedded within the HTML of a web document. When a user requests the document, the PHP script runs on the server-side, and performs any programs and operations specified in the script. A dynamically generated HTML page is then delivered to the user.

PHP runs as an optional module within the Apache web server. It is an open source alternative to Microsoft's Active Server Pages (ASP), Sun's Java Server Pages (JSP), and the like. Originally developed in 1994, the first version of the program was known as "Personal Home Page Tools."

Python

A script programming language noted for its portability and clear syntax. It is an object-oriented language that incorporates modules, exceptions, dynamic typing, very high level dynamic data types, and classes. Extensions to Python can be written in the C or C++ programming languages. It is available under an open source license.

RDBMS (Relational Database Management System)

A collection of programs that allows you to create, store, modify, and administer a relational database. An RDBMS stores data in related tables, and information can be extracted from the database through structured query language (SQL) statements.

Because the data in a relational system is spread across tables, rather than housed in a flat file, the same database can be viewed in many different ways. Almost all complex databases today use an RDBMS, including most business databases.

RIA (Rich Internet Application)

A term for web applications that have the functionality and features of traditional desktop applications. Typically, the applications transfer necessary functions to the client (in this case, the web browser), which enables the avoidance of a page having to refresh every time a new piece of information is needed. While RIAs run in a web browser, they don't usually require software installation.

REST

REST stands for "Representational State Transfer," and is a software architecture for distributed internet systems. Specifically, it is an alternative to Web Services and SOAP for integrating services and repositories without requiring messaging or cookies.

RSS

Short for "really simple syndication," RSS is a very straightforward XML model for syndicating headlines and descriptions to other websites or RSS readers.

SOAP

SOAP is the predominant standard protocol in the Web Services family. It is an XML construct that allows applications to be invoked remotely and deliver information back to the calling service.

Taxonomy

In science, taxonomy allows people to identify any organism by its kingdom, phylum, class, order, family, genus, and species. Taxonomy, as it relates to content management, does the same job. It describes a classification structure for content. This structure, typically highly regimented, affects the data model, directory structure, and file naming conventions for a given implementation of a content management system. In more complex scenarios, taxonomies are often multi-faceted, meaning multiple hierarchies or categorization trees may be used to classify content. This allows users to find content via more than one path or hierarchy (as an example, one might find information about red rock crabs via a biology facet under animals / invertebrates / crustaceans, while another might find one via a geography facet under world / land / Australasia). Taxonomy can also be language-oriented, as in specifications for subsets of XML, such as ebXML.

Thesaurus

A type of controlled vocabulary that shows the hierarchical (e.g., parent-child), associative (e.g., related) and equivalent (e.g., synonymous) relationships among terms.

Visual Basic (VB)

A programming environment developed by Microsoft that provides a graphical interface for writing code in the BASIC (Beginner's All-Purpose Symbolic Instruction Code) language. VB allows programmers to select and modify bits of existing BASIC code, and assemble them into custom applications and programs.

VB is particularly suited to rapid application development, and therefore is often used for prototyping applications that might subsequently be developed in a more powerful language like C or Java. VBScript is a subset of the language that is frequently used to apply dynamic logic within Active Server Pages.

WebDAV

“Web-based Distributed Authoring and Versioning”. It is a set of extensions to the HTTP protocol which allows users to edit and manage files on remote web servers collaboratively. (From www.webdav.org) As a practical matter, in a CMS it can allow authors to place and retrieve remote files in/from a WebDAV-compatible repository using Windows Explorer or WebDAV-compatible desktop tool.

Wire Frames

A rough outline of page elements and their arrangement within the page. An important deliverable in any template or site redesign, wire frames show the logical elements on the page, as opposed to the visual elements of a page.

WML (Wireless Markup Language)

The markup language used to format text and data for wireless application protocol (WAP) devices, including smartphones. WML is an extensible markup language (XML) application.

XML eXtensible Markup Language

XML is a markup language created so that richly structured documents can be described, exposed, shared, and modified over the web.

Since XML describes the underlying information and its structure, content can be separated from look-and-feel. This overcomes a severe limitation of HTML, which merely describes content presentation for a particular set of HTML-compliant applications (like web browsers).

Vendor Index

A

Adobe
Compare to 240, 243, 273, 286, 330, 357,
407, 481, 540
Main evaluation 248–272

C

CCI Escenic
Compare to 478, 481, 483, 528, 540, 612
Main evaluation 648–661

Clickability 5
Compare to 478, 490

CoreMedia 4, 139, 239, 240, 245, 247
Compare to 147, 195, 219, 478, 481, 483,
528, 540, 648
Main evaluation 273–285

CrownPeak 5, 140
Compare to 483, 612, 667, 747, 778
Main evaluation 490–507

D

DotNetNuke 6, 141
Compare to 664, 669, 709, 789, 815
Main evaluation 673–692

Drupal 4
Compare to 357, 359, 407, 443, 460, 481,
483, 557, 612, 664, 669, 693, 730,
815
Main evaluation 372–384

E

Ektron 5
Compare to 243, 330, 356, 357, 386, 423,
478, 481, 483, 490, 557, 578, 592,
612, 664, 669, 673, 709, 789
Main evaluation 508–527

Enonic 5
Main evaluation 528–539

EPiServer 5, 140
Compare to 243, 330, 478, 481, 483, 508,
528, 557, 592, 664, 709
Main evaluation 386–406

e-Spirit
Compare to 478, 483, 528, 633
Main evaluation 540–556

eZ Systems 5
Compare to 359, 443, 460, 483, 633, 664, 693
Main evaluation 557–577

H

Hannon Hill 5
Compare to 483, 592, 667, 669, 747, 801
Main evaluation 578–591

Hippo 5
Compare to 483, 633, 667, 764
Main evaluation 407–422

HP
Compare to 243, 286, 307
Main evaluation 151–173

I

IBM 4, 139, 145, 148, 150
Compare to 147, 195
Main evaluation 174–194

Ingeniux 5
Compare to 481, 578, 667, 747
Main evaluation 592–611

J

Joomla! 6
Compare to 356, 372, 664, 673, 730
Main evaluation 693–708

K

Kentico
Compare to 356, 357, 386, 423, 478, 508,
664, 669, 673, 789
Main evaluation 709–729

M

Magnolia 5
Compare to 240, 273, 357, 407, 478, 481,
483, 528, 540, 648, 667, 764
Main evaluation 633–647

Microsoft 5

Compare to 243, 330, 356, 386, 483, 592,
664, 673

Main evaluation 423–438

MODX

Main evaluation 730–746

O

OmniUpdate 6

Compare to 483, 612, 669, 801

Main evaluation 747–763

Open Text Vignette WEM

Compare to 145, 151, 240, 243, 248, 286,
481, 578

Main evaluation 195–218

OpenCms

Compare to 357, 359, 407, 460, 481, 578

Main evaluation 764–777

Oracle WebCenter/FatWire

Compare to 145, 147, 151, 174, 195, 240,
243, 248, 286, 307, 478, 528

Main evaluation 219–238

P

Percussion 4, 139, 242, 245, 247

Compare to 669, 801

Main evaluation 286–306

Plone 5

Compare to 356, 359, 372, 460, 481, 557,
664, 667, 673, 730, 764

Main evaluation 443–459

Polopoly Atex

Compare to 483, 648

Main evaluation 363–371

S

Salesforce

Main evaluation 778–788

SDL Tridion 4, 139, 242, 245, 247

Compare to 145, 174, 240, 243, 248, 273,
286, 330, 356, 386

Main evaluation 307–329

Sitecore 5

Compare to 240, 243, 248, 307, 356, 357,
386, 423, 478, 483, 490, 508, 592,
612, 664, 669, 709, 789

Main evaluation 330–353

T

Telerik 482

Compare to 356, 357, 386, 423, 478, 508,
664, 709

Main evaluation 789–800

TERMINALFOUR

Compare to 669, 815

Main evaluation 801–814

TYPO3 354

Compare to 356, 372, 481, 557, 664, 693

Main evaluation 460–475

U

Upland Software

Compare to 478, 490

Main evaluation 612–632

W

WordPress

Compare to 356, 372, 664, 667, 673, 730, 778

Main evaluation 815–832