Lesson 2

Make View: Check Code

SUMMARY

This lesson is an introduction to creating Check Code inside the Make View module of Epi Info™. In this lesson, you will learn how to customize your survey by creating a series of data entry rules. You will also learn the components of the Make View Program Editor.

In Lesson 1, you created a variety of fields to collect data for your survey. In Lesson 2, you will use the survey you created and add Check Code to some of the fields. Check Code makes the data entry process faster and more efficient, so there are fewer errors when you begin using the data later to create the statistics and develop your report.

Time to complete: 2 hours Intermediate

Getting Started with Check Code

Overview of basic information to navigate the Program Editor and understand Check Code. A list of available commands is provided. Functions and Operators are defined.

Create a Skip Pattern

Use the Program Editor to create an IF/THEN statement. Also introduces the commands ELSE and GOTO.

Use the ASSIGN Command

Use the Program Editor to create a mathematical function that calculates the Age field based on entries in the Date of Birth field. Use the YEARS function.

Use the DIALOG Command

Use the Program Editor to notify data entry personnel of specific data entry rules or errors to avoid.

Skills Review

Series of five review questions based on the reading and hands-on activities in this lesson.

BEFORE YOU BEGIN

Complete Lesson 1 Make View: Create a Survey

WHAT YOU NEED

Appendix A: Code Sheet

Appendix B: Parent School Asthma Pre-Intervention Survey

Asthma Survey.MDB- created in Lesson 1

Appendix D: Skills Reviews Answer Key

FIVE GOALS

- Be able to open and navigate the three sections of the Make View Program Editor: the Command Tree, the Check Commands window, and the Program Editor code window.
- Create a skip pattern using an If/Then statement.
- Create code using the ASSIGN command.
- Create code to calculate age using the YEARS function.
- Create a DIALOG box that appears during data entry and prompts the user to verify their data.

Getting Started with Check Code

Check Code is used to validate data entry and to make the process of entering data faster. With some advance planning, you can create code that will do calculations, skip questions based on answers, prompt the user with dialog boxes, and populate fields across pages and records. In its most basic form, Check Code is a set of rules for the person entering data to follow. Check Code helps to eliminate errors that can occur when entering large amounts of data.

Open the Program Editor by selecting the blue Program button located in the Page Names window or by selecting **View>Check Code** from the Make View navigation menu.

The Make/Edit View: Check Commands window contains three working areas: the Check Command Tree, the Make/Edit View: Check Commands window, and the Program Editor. The Check Command Tree is not visible until you select a variable from the drop-down menu.

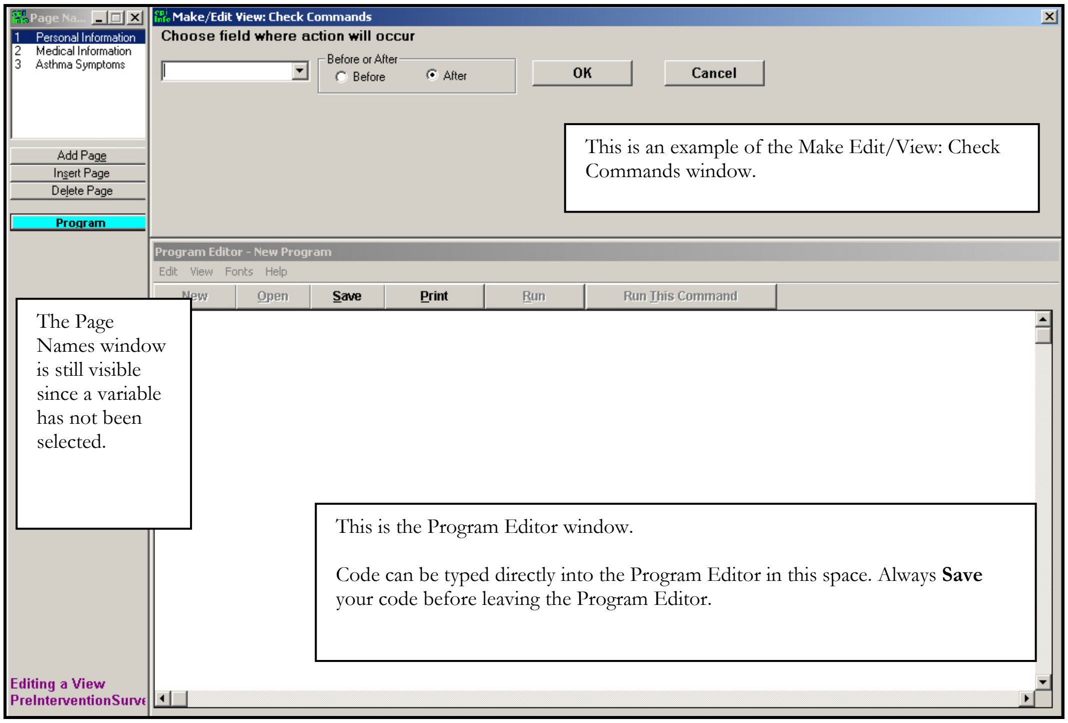
The Program Editor Workspace

To access the Program Editor and create Check Code, you must open the survey you created in Make View Lesson 1.

1. Open the Program Editor by clicking the blue **Program** button from the Page Names window or selecting **View>Check Code**. The Make/Edit View Check Commands window opens.

   - Notice that the Page Names window is still open since a variable has not been selected.

   This is an example of the Program Editor/Check Code workspace. The top of the page shows you are in the Make/Edit View: Check Commands window.



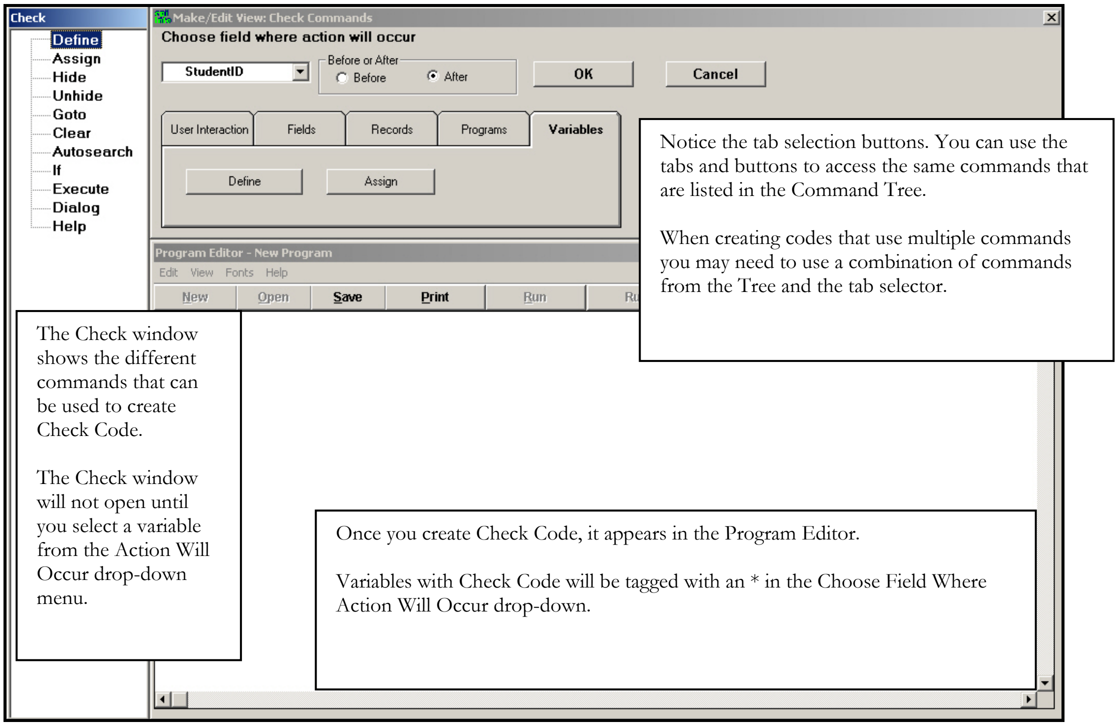
This is an example of the Make Edit/View: Check Commands window.

The Page Names window is still visible since a variable has not been selected.

This is the Program Editor window.

Code can be typed directly into the Program Editor in this space. Always **Save** your code before leaving the Program Editor.

2. From the Choose Field Where Action Will Occur drop-down, select a variable. The Make/Edit Check Commands window updates to include the Check Command Tree and the tab selector.



Choose field where action will occur: StudentID

Before or After: ○ Before ● After

The Check window shows the different commands that can be used to create Check Code.

The Check window will not open until you select a variable from the Action Will Occur drop-down menu.

Notice the tab selection buttons. You can use the tabs and buttons to access the same commands that are listed in the Command Tree.

When creating codes that use multiple commands you may need to use a combination of commands from the Tree and the tab selector.

Once you create Check Code, it appears in the Program Editor.

Variables with Check Code will be tagged with an * in the Choose Field Where Action Will Occur drop-down.

- Check Code must be associated with an existing variable, page, view, or record. The commands are executed when the data entry cursor enters or leaves the selected variable. Variables available for Check Code are organized by page in the drop-down menu.

- The Check window and Command Tree display a list of all available check commands that can be used in the Make View program. Clicking on one of the Commands opens the wizard or dialog box needed to create the selected kind of code.

- The Make/Edit View: Check Commands window allows you to select when and to which variable the Check Code program you create will be applied. Previously created Check Code can be accessed from this window. Use the drop-down list to view the available variables. Any variable tagged with an asterisk (*) has Check Code associated with it. Select the variable to view the code inside the Program Editor. Since you have not created code yet, none of your variables will be tagged.

- The Program Editor displays the commands created from the Check Command Tree. If you have experience with coding, you can type directly into the Program Editor and save your code without using the wizards or the Command Tree.

Navigating the Program Editor

Familiarize yourself with the available commands and options present in the Program Editor.

- Click **Cancel** to close any Command Option boxes you open during the Overview.
- Click **Cancel** from the Check Commands window or the close X to exit the Program Editor prior to starting the next lesson.

Make/Edit View: Check Commands Window

The **Choose Field Where Action Will Occur** drop-down shows a list of fields in the current view. Read Only variables will not execute Check Code.

The Before or After radio buttons tell when the action will occur or when your Check Code will run in the data entry process. You can create code that runs before you enter data in the selected field or after you enter data when your cursor leaves the field.

The Tab section gives you another way to navigate and select commands. Click a tab to open it and see which command buttons are associated with it. Commands selected from the Tab section open the same wizards/generators and dialog boxes as the Command Tree.

Check Window

The Command Tree is located inside the Check window. Commands located inside the Check window act as buttons when clicked. The dialog boxes that open when you select a command are also called wizards or generators as they are used to guide you through the Check Code process.

Once you begin creating code, the Command Tree will not be active while inside some of the dialog boxes. When that occurs you must use the tabs to select commands.

Available Commands

Define

Use this command to create new variables. Located on the Variables Tab.

Assign

Use this command to assign the result of arithmetic or string expression mathematics to a variable. Located on the Variables Tab.

Hide

Use this command to hide a field from view and make it a read only (NOENTER) field. Located on the Fields Tab.

Unhide

Use this command to make a field visible and return it to the status it had before it was hidden. Located on the Fields Tab.

GoTo

Use this command alone or in an IF statement to transfer the cursor to a specifically named variable field. Located on the Fields Tab.

Clear

Use this command to set the field named to the missing value, as though the field had been left blank. CLEAR is frequently followed by the GOTO command that puts the cursor in position for further entry after an error. Located on the Fields Tab.

Autosearch

Use this command to cause Enter to search for one or more matching records. If a match is found, you can display and edit the matching record(s) or ignore the match and continue to enter the current record. Located on the Records Tab.

If

Use this command to define conditions and/or consequences that result when the conditions are met. An alternative consequence can be given after the ELSE statement. Located on the Records Tab.

Execute

Use this command to execute a Windows or DOS program. Located on the Programs Tab.

Dialog

Use this command to provide interaction with users from within a program. Dialogs can display information, ask for and receive input, and offer lists for making choices. Located on the User Interaction Tab.

Help

Use this command to display an .HTM or .CHM document. Located on the User Interaction Tab.

Program Editor Menu

Edit

Use the Edit Menu to copy, cut, and paste code. You can also edit your code with Delete Line and Replace Line options. Search your code using Find, Find Next, Program Beginning and Program End options.

View

Use the View Menu to see additional menu options. Use the Status Bar to see information displayed about your script. Use the Tool Bar to edit your script.

Fonts

Use the Fonts Menu to change the font type, style, size, and color that appears in the Program Editor.

Help

Use the Help menu to view the help file contents.

Save

Use the Save button to save the Check Code you created in the Program Editor. The Save button also verifies that the code syntax is correct.

Print

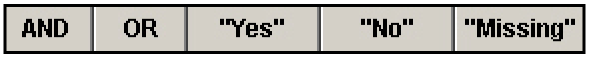
Use the Print button to make a paper copy of your code.

Functions and Operators

Check Code is created using a series of functions and operators. Functions and Operators are used inside commands to perform tasks such as extracting a year from a date, combining two numeric values, or testing logical conditions. Check Code can be typed directly into the Program Editor or created using the dialog box generators. Within each dialog box, there are Function and Operator buttons to guide you in creating the proper syntax for your Check Code. You can also click **Functions** from within each dialog box to see a list of all the available functions and examples of the proper syntax.

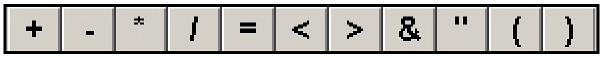
Functions modify the value of one or more variables to produce a result. For example, the function ROUND produces a whole from a variable that has decimal places. Almost all functions require arguments enclosed in parentheses and separated by commas. Where arguments are required, do not place spaces between the function name and the left parenthesis.

These are the function buttons available using the Program Editor dialog boxes.



Operators are used to combine two items. For example, the "+" operator combines Var1 and Var2 to produce a sum, as in Var3=Var1+Var2.

These are the operator buttons available using the Program Editor dialog boxes.



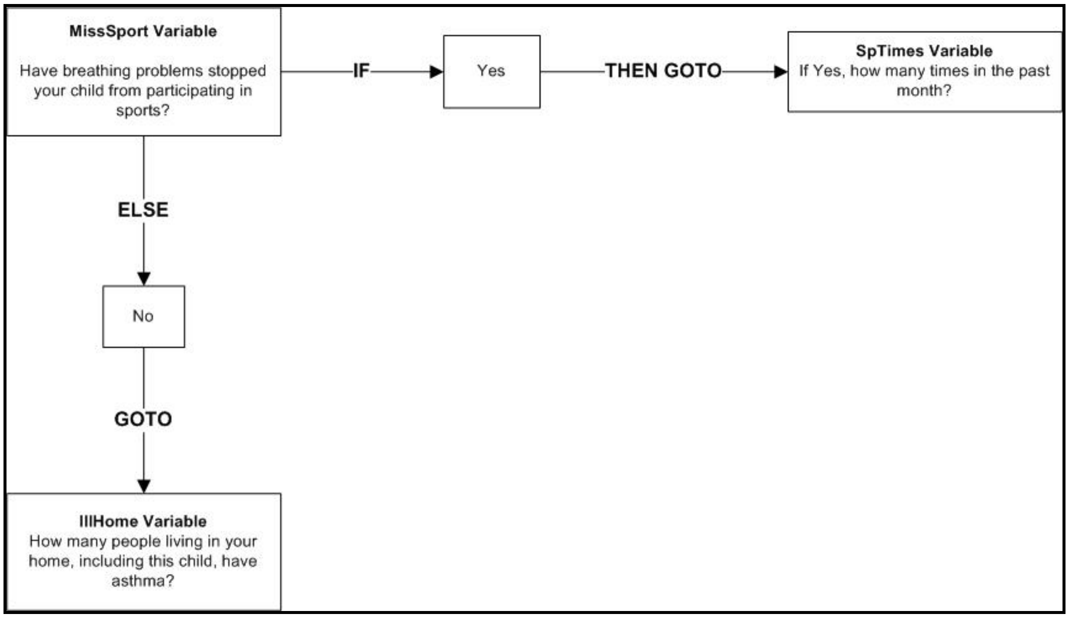
+ Addition	= Equal or Assign
- Subtraction	< Less Than
* Multiplication	> Greater Than
/ Division	<> Less Than or Greater Than (not Equal to)
& String Concatenation	" Enclose dialog or text
(Parenthetical Expression	) Parenthetical Expression

Create a Skip Pattern

Skip patterns can speed up the data entry process by moving the cursor to a specified field based on the answer to a question. In this survey, parents were asked if their children have missed sports due to breathing problems. If they have missed sports, then you want them to specify how often, if they have not then you want them to move to the next question.

To create a skip pattern, you will need to use the IF command. First, plan out what you want the code to accomplish.

If a respondent answers **Yes**, they have missed sports, **Then** you want the cursor to **GoTo** the next question. **Else** (if they answer No) you want the cursor to **GoTo** (or skip) to a new question.


MissSport Variable
Have breathing problems stopped your child from participating in sports?
IF
Yes
THEN GOTO
SpTimes Variable
If Yes, how many times in the past month?
ELSE
No
GOTO
IllHome Variable
How many people living in your home, including this child, have asthma?

IF/THEN Diagram: Creating a Skip Pattern

To create a skip pattern inside the survey, you are going to use the Program Editor.

1. Click **Program** from the Page Names window. The Make/Edit View: Check Commands window opens.

2. From the Choose Field Where Action Will Occur drop-down, select MissSport.

   - You want the action to occur After data is entered into the MissSport field.

3. From the Records tab or the Check Command Tree, select **If**. The IF dialog box opens.

4. From the Available Variables drop-down, select MissSport. The selected variable appears in the If Condition field.

5. From the Operators, click **=**.

6. From the Operators, click **"Yes"**.

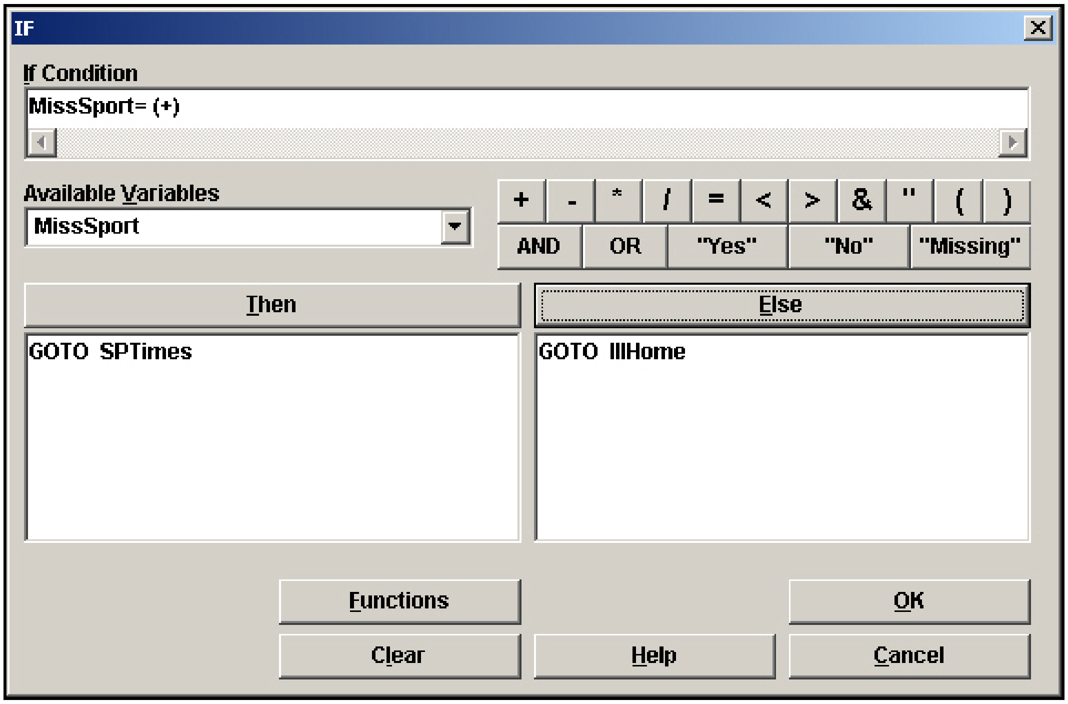
    - The If Condition field will read MissSport=(+)

7. Click **Then**. The tab selector appears. The Check Command Tree is not active.

8. From the Fields tab, select **GoTo**. The GOTO dialog box opens.

9. Select SPTimes from the list of variables.

10. In the GOTO dialog box, click **OK**. The IF dialog box opens.



IF

If Condition

MissSport= (+)

Available Variables

MissSport

Then

GOTO SPTimes

Else

GOTO IllHome

11. Click **Else**. The dialog box disappears.

12. From the Fields tab, select **GOTO**. The GOTO dialog box opens.

13. Select IllHome from the list of variables.

14. Click **OK**. The IF dialog box opens.

15. Click **OK**. The code appears in the Program Editor:

```
IF MissSport= (+) THEN
    GOTO SPTimes
ELSE
    GOTO IllHome
```

END

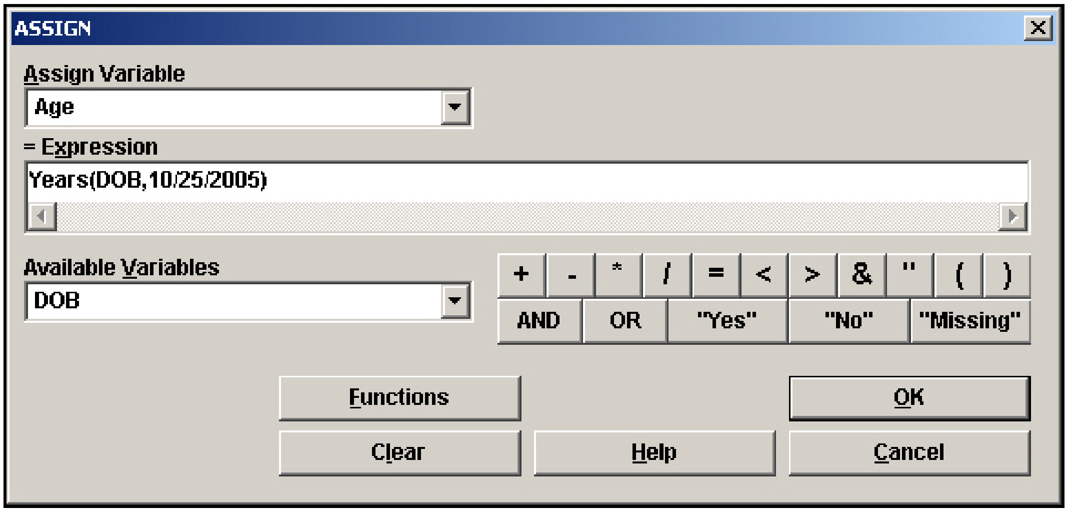
16. Click **Save**.

- Notice in the drop-down that the field MissSport now has an asterisk meaning there is code associated with that field.

Use the ASSIGN Command

To program a mathematical function, you must use the Program Editor. In the survey, you have a field called Date of Birth and a field called Age. You are going to create Check Code that will calculate and enter the age of the respondent based on the date of birth and the date the survey was completed.

1. From your View, click **Program** or select **View>Check Code**. The Program Editor opens.
2. From the Choose Field Where Action Will Occur drop-down, select DOB. The Check Command Tree opens.
   - **After** is the default setting for when all actions will occur. The calculation needs to run after you enter information into the Date of Birth field.
3. From the Variables tab or the Check Command Tree, click **Assign**. The ASSIGN dialog box opens.
   - To complete the code, use the Functions and Operators buttons and the keyboard to type in the = Expression field.
4. From the Assign Variable drop-down, select Age.
   - The calculated value appears in the Age field.
5. In the = Expression field, type the function *YEARS*.
   - To see a list of all the available functions, click **Functions**.
6. Type, or click, the left parenthesis.
   - Statements of a function must be enclosed in parentheses. Use the Operator buttons or type them in from your keyboard.
7. From the Available Variables drop-down, select DOB.
8. Type a comma.
9. Type the survey date of 10/25/2005.
10. Type or click the right parenthesis.



Assign Variable
Age

= Expression
Years(DOB,10/25/2005)

Available Variables
DOB

11. Click **OK**. The Check Code appears in the Program Editor.

    - ASSIGN Age=Years(DOB,10/25/2005)

12. Click **Save** in the Program Editor.

    - Always save your Check Code. Codes will not update unless you save. The Save feature will also inform you of any syntax errors that will affect the code functionality.

    - Notice that DOB now has an asterisk next to it in the Choose Field Where Action Will Occur drop-down menu. This signifies that the variable has Check Code associated with it.

13. Click **OK** in the Check Commands window to return to the Make/Edit View main page.

Use the DIALOG Command

You are going to create Check Code that will provide a dialog box and added instructions for the person entering data into the project survey. The DIALOG command provides interaction with the data entry personnel from within a program. Dialogs can display information, ask for and receive input, and offer lists for making choices.

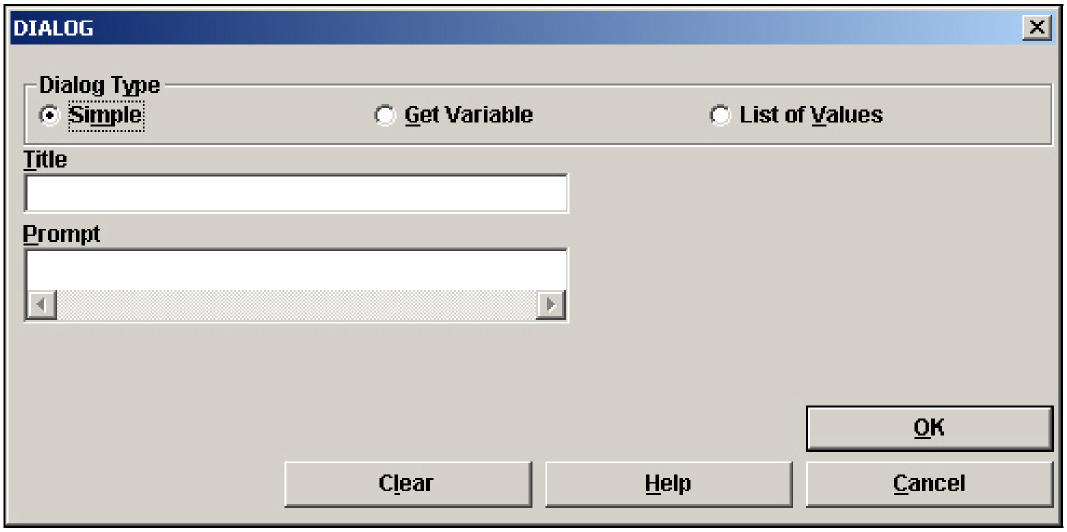
In the survey, you have a variable called MissDays. This variable refers to the question "How many days of school has your child missed due to asthma?" You want to make sure that any half days are rounded up and counted as whole days during the data entry process. Using the DIALOG command allows you to set a reminder to run during data entry. This will help to ensure you get the most accurate count from your survey data and that all persons entering data will be doing so based on the same set of rules.

1. From your View, click **Program** or select **View>Check Code**. The Program Editor opens.

2. From the Choose Field Where Action Will Occur drop-down, select MissDays.

   - You want the action to occur after data is entered into this field.

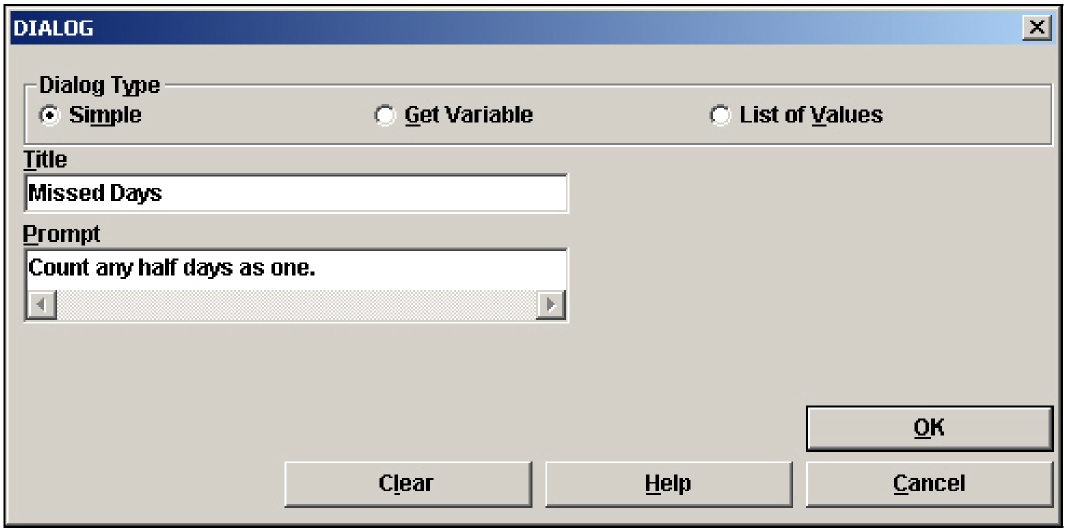
3. From the User Interaction tab or the Check Command Tree, select Dialog. The DIALOG box opens.



4. In the Title field, type *Missed Days*.

   - The Dialog Type radio button will be selected as Simple.

5. In the Prompt field, type *Count any half days as one.*



DIALOG

Dialog Type

● Simple ○ Get Variable ○ List of Values

Title: Missed Days

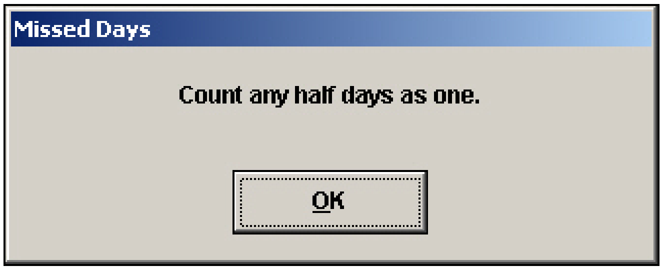
Prompt: Count any half days as one.

6. Click **OK**. The code appears in the Program Editor.

   DIALOG "Count any half days as one." TITLETEXT="Missed Days"

7. Click **Save** in the Program Editor.

   - When you test your code in the next chapter on Enter Data the following dialog box should appear once you enter data in the MissDays variable.



Missed Days

Count any half days as one.

🕮 Skills Review

Place a checkmark by the answer to the following questions. Refer to *Appendix D Skills Reviews Answer Key* for additional details and the answers.

1. Check Code can run Before or After values are entered in a variable.

   □True

   □False

2. Using the Program Editor, you can create Check Code that will run across pages or records.

   □True

   □False

3. In the Check Command window, variables tagged with an X have Check Code associated with them.

   □True

   □False

4. Which Function is used to create an age variable?

   □ROUND

   □YEARS

   □TXTTONUM

5. How does the answer Yes appear in the Program Editor?

   □(+)

   □(-)

Lesson Complete!

WHAT YOU LEARNED

How to

- Navigate the Program Editor.
- Create a Skip Pattern using the IF/THEN commands.
- Use the ASSIGN Command.
- Use the YEARS function.
- Use the DIALOG command.