Checking data entry

Introduction

Check Code validates data entry and enters data faster and more accurately. With advance planning, code can be created to perform calculations, skip questions based on prior answers, prompt the user with dialog boxes, and populate fields across pages and records. Basically, Check Code is a set of rules for you to follow while entering data. It helps eliminate errors that can occur when you enter large amounts of information. Check Code is created using the Check Code Editor.

Check Commands must be placed in a block of commands corresponding to a variable in the database. Commands are executed before or after you display a form, page, or record. Before commands can be inserted into the Check Code Editor, a Check Code Block corresponding to a form, page, record, or field can be created.

To create Check Code, open the Program Editor by selecting the **Check Code button** located on the toolbar, or select **Tools > Check Code Editor** from the Form Designer navigation menu. Check Code Blocks are created in the following way:

1. Select the form, page, record, or field that will receive the commands.
2. Select **before** if the commands will be executed before data entry into the form, page, record, or field. Otherwise, select **after** if the commands will be executed after data entry when the cursor leaves the form, page, record, or field.

EXERCISES

Create a Skip Pattern Using IF/THEN and GOTO

Use IF/THEN statements to create skip patterns based on the answers to questions in the survey or questionnaire. 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.

1. From the Epi Info main menu, select **Create Forms** or select **Tools > Create Forms**. The Form Designer window opens.
2. Select **File > Recent Projects > C:\RMS Course\AsthmaSurvey\AsthmaSurvey.prj** to open the AsthmaSurvey project we created in the last lesson.
3. **Click Check Code**. The Check Code Editor opens.
4. From the **Choose Field Block for Action** list box, select the field which contains the action. Select **MissSport**.
   - The action needs to occur after data are entered into the MissSport field. Select **after** from the Before or After section.
5. Click the **Add Block** button. This creates code to run after the field is entered and accepted.
6. Click **If** from the **Add Command to Field Block** list box. The IF dialog box opens.

7. From the Available Variables drop-down list, select the **MissSport** field to contain the action.
8. From the Operators, click =.
9. From the Operators, click **Yes**. The If Condition field will read MissSport=(+).
10. Click the **Code Snippet** ( ) button in the Then section. A list of available commands appears.
11. From the command list, select GoTo. The GOTO dialog box opens.
12. Select the **SPTimes** field for the cursor to jump to based on a Yes answer from MissSport.
13. In the GOTO dialog box, click **OK** to return to the IF dialog box.
14. Click the **Code Snippet** button in the Else section. A list of available commands appears.
15. From the command list, select **GoTo**. The GOTO dialog box opens.
16. Select the **IllHome** field for the cursor to jump to based on a No answer from MissSport.
17. In the GOTO dialog box, click **OK** to return to the IF dialog box.
18. Click OK. The code appears in the Check Code Editor as:

```
Field MissSport
 After
        //add code here
        IF MissSport = (+)  THEN
                GOTO SPTimes
        ELSE
                GOTO IllHome
        END-IF
 End-After
End-Field
```

19. Click **Save** from the Check Code Editor. Always save Check Code. Code will not update unless saved. The Save feature also verifies syntax.

Use the ASSIGN Command

Check Code can be created to calculate and enter the age of a respondent based on the date of birth and the date the survey was completed, or the system date of the computer when data were entered.

This example uses the **DOB** field and the **Age** field to demonstrate the use of the **ASSIGN** command and the function **YEARS.**

1. From the Form Designer, click **Check Code** or select **Tools > Check Code Editor.** The Check Code Editor opens.
2. Select the **DOB** field from the Choose Field Block for Action list box.
3. Select **after** from the Before or After Section.
4. Click the **Add Block** button. This creates code to run after the Child's Date of Birth field is entered and accepted.
5. Click **Assign** from the Add Command to Field Block list box. The Assign dialog box opens.
6. From the Assign Variable drop-down, select the **Age** field where the calculated value should appear.
7. In the = Expression field, type the function **YEARS** or click the function ***f(x)*** button to the right of the Expression box and select **Date Functions** > **YEARS**.
8. Type or click the left parenthesis **(**.

- Do not put a space before it. Statements of a function must be enclosed in parentheses. Use the Operator buttons or type them in from your keyboard.

9. From the Available Variables drop-down list, select the **DOB** field. Type a **comma**.
10. Type the **survey date** of **17/02/2014** in a DD/MM/YYYY format[1] or type the function **SYSTEMDATE** to calculate using the computer clock.
11. Type or click the right parenthesis **)**.
12. Click **OK**. The Check Code can appear in the Check Code Editor in two ways.

```
ASSIGN Age = YEARS(DOB, 17/02/2014)
ASSIGN Age = YEARS(DOB, SYSTEMDATE)
```

13. Click the **Save** button in the Check Code Editor.
    - Always save Check Code. Code will not update unless saved. The Save feature also verifies syntax.

Create a Dialog

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 the form, click **Check Code** or select **Tools > Check Code Editor**. The Check Code Editor opens.
2. From the Choose Field Block for Action list box, select MissDays
3. Select **after** from the Before or After Section. Click the **Add Block** button.
4. From the Add Command to Field Block list box, select **Dialog**. The DIALOG box opens.
5. In the Title field, type *Missed Days*. The Dialog Type radio button should be Simple.
6. In the Prompt field, type *Count any half days as one*.
7. Click **OK**. The code appears in the Check Code Editor.

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

8. Click **Save** in the Check Code Editor.

[1] Or MM/DD/YYYY if your PC is using a US locale