Epi Info™ 7 User Guide

Version 3

3/21/2016

User Guide Chapters

1. Introduction to Epi Info™ 7

Using Epi Info™

Epi Info™ 7 is a series of tools designed to help public health professionals conduct outbreak investigations, manage surveillance databases, and perform statistical analyses. The software enables epidemiologists and other public health and medical professionals to createa questionnaire , customize the data entry process, and enter and analyze data. The program is free and publicly available for download at the Epi Info™ website (http://wwwn.cdc.gov/epiinfo/). The software runs on Microsoft Windows operating systems. Download and installation information is located in the appendix of this guide.

Tools

Epi Info™ 7 is comprised of four main tools used to collect, analyze, or visualize data:

- **Create Forms** – Create a questionnaire using one or more forms to collect and view data.
- **Enter Data** – Enter data and view existing records.
- **Analyze Data (Classic** and **Visual Dashboard)** – Manage data, run statistical analyses, and generate lists, tables, graphs, and charts.
- **Create Maps** – Create maps from map server, KLM files, or shape files.

Available utilities include:

- **StatCalc** – Compute statistics from summary data.
- **Help** – Find online and offline resources: including a “discussion forum” and “help desk” information.
- **Options** – Configure default settings on the following tabs: general, language, analysis, plug-ins, and web survey.

Epi Info™ 7 comes with several fully functional sample projects that can be used as templates when designing forms. The Nutrition project (nutrition.prj) contains the functionality of NutStat also found in previous version of Epi Info™. It contains several nutritional anthropometry functions that can be used to collect, analyze, and graph child growth data. Other sample projects contain various forms that demonstrate the potential use of Epi Info 7™ in public health data collection and outbreak investigations.

Conventions Used in the User Guide

This following table identifies typographic conventions used in this document.

Convention	Description
Boldface type	Emphasizes heading levels, column headings, and the following text when writing procedures: • Names of options and elements that appear on screens if the option or element is involved in an action • Keys on the keyboard • User input for procedures • Definitions
Italic type	Accentuates words or phrases that have special meaning or that are being defined.
`Courier New`	Used for code samples.
Hyperlink or URL	Hyperlinks and URLs are highlighted in blue and may be underlined.
File > Print	Used to identify menu choice and command selection.

Syntax Notations

The following conventions/rules apply to this document.

Syntax	Explanation
ALL CAPITALS	Epi Info 7™ commands and reserved words are shown in all capital letters.
<parameter>	Information to be supplied to a command or function. Parameters are enclosed with less-than and greater-than symbols. Each parameter is described following the statement of syntax for the command. Parameters are required by the command unless they are enclosed in braces { }. Do not include the < > or { } symbols in the code.
[<variable(s)>]	Brackets [ ] around a parameter indicate the possibility of more than one parameter.
{<parameter>}	Braces { } around a parameter indicate an optional parameter. Do not include the < > or { } symbols in the code.
|	The pipe symbol '|' denotes a choice and is usually used with optional parameters. An example is seen in the ROUTEOUT command. `ROUTEOUT <filename> {REPLACE | APPEND}`
*	An asterisk in the beginning of a line of code, as shown in some code examples, indicates a comment. Comments are skipped when a program is run. An asterisk also serves as "all" when coding in Classic Analysis.
" "	Quotation marks must surround all text values. `DIALOG "Notice: Date of birth is invalid."`

System Requirements

These requirements apply to versions available for download up to and including Epi Info™ 7.1.1.14.

- Microsoft Windows XP or above
- Microsoft .NET Framework 4.0 or above
- Recommended – 1 GHz processor
- Recommended – 256 MB RAM

DOWNLOAD NOTES

Epi Info™ 7 is available for download in two different formats, as a setup or zip file. The following explains what scenarios may be best suited for each format.

Setup (.exe file) Installation

- Traditional setup mechanism that deploys Epi Info™ 7 (If applicable, the location is subject to an organization's IT policy).
- Allows network administrators to centrally manage and push Epi Info™ 7, including updates and patches, to users using Microsoft System Center Configuration Manager.
- Ensures that the desktop configuration matches the software's minimum requirements.
- Precompiles and registers Epi Info™ 7 components on the computer, boosting the speed of certain components.
- Requires administrative or elevated privileges during installation.
- Recommended for centrally managed IT environments.

ZIP (.zip file) Installation

- Can be downloaded to most computers and run without requiring administrative or elevated privileges.
- Can be extracted to and run from any folder for which the user has read/write/execute privileges (including thumb drives).
- Assumes that the computer already has Microsoft .NET 4.0 and other prerequisites installed.

- Recommended for disconnected laptops and other emergency use if IT support or infrastructure is unavailable.

Navigate Epi Info™ 7

Epi Info™ 7 tools can be accessed through the main menu. Open the main menu by double-clicking the Epi Info™ icon on the desktop.

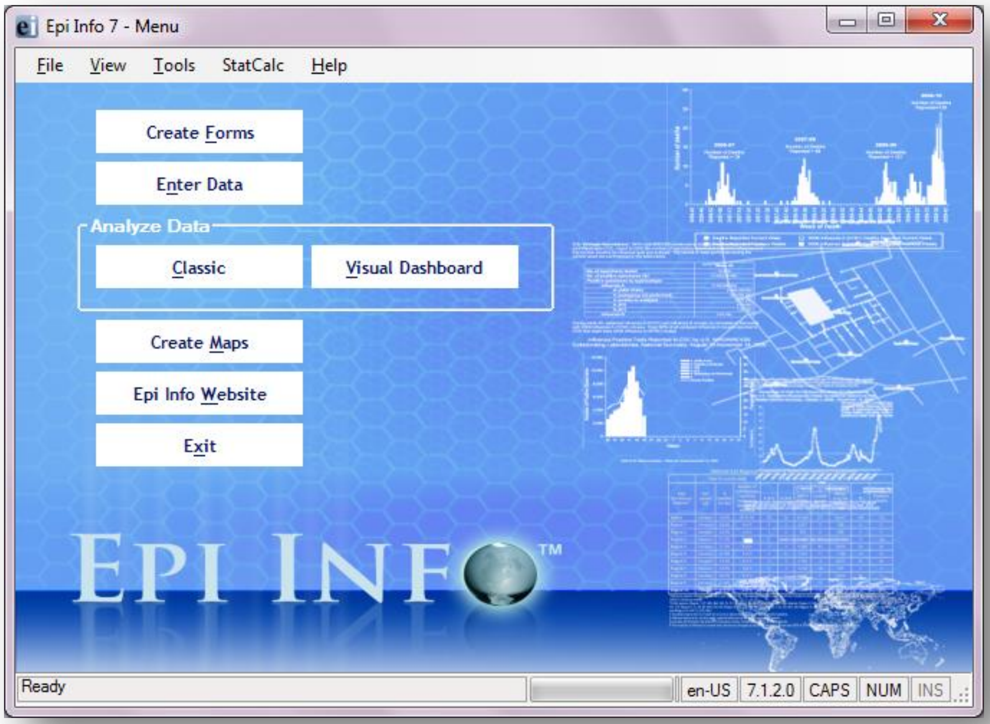


Figure 7.1: Epi Info™ Main Menu

1. The navigation menu, located at the top of the Epi Info™ main menu, allows access to tools and setting customization.

- **File** allows you to exit Epi Info™ 7.
- **View** offers two display options**.**
  - **Status Bar** displays several items at the bottom of the screen, including the program version number. The version number will vary based on the version downloaded from the Epi Info™ website. Versions of Epi Info™ may also display the release date.
  - **Epi Info™ Logs** keep a record of certain program errors and activities.
- **Tools** allows access to **Create Forms, Enter Data, Analyze Data (Classic** and **Visual Dashboard)**, and **Create Maps**.
    - **Options** enable you to set default options. Detailed descriptions of these options are located in the appendix.

    - **General** – includes those for the background image and default data formats.
    - **Language** – allows file translation into multiple languages. For additional information on language options, reference the Translation section of the user guide.
    - **Analysis** – provides for options on the display of Boolean values, HTML output, and statistical functions.
    - **Plug-ins** – allows the user to import data sources and custom designed dashboard gadgets.
    - **Date\Time** –allows the user to set formats for date, time, and date and time as a single unit (***Note: not available in all versions of Epi Info™)***.
    - **Web Survey** – enables the user to identify an endpoint address and set various survey protocols. For additional information on web survey options, reference the Web Survey section of the user guide.
- **StatCalc** is a statistical calculator used to compute statistics from summary data. It is frequently used to calculate sample size, chi square test, relative risks, and odds ratios.
- **Help** provides access to the Epi Info™ User Guide, online help videos, an Epi Info™ discussion forum, and instructions on how to contact the help desk.

2. Menu options are located on the left side of the screen. These options allow easy access to the most used tools: **Create Forms**, **Enter Data**, **Analyze Data**: **Classic** and **Visual Dashboard**, **Create Maps**, **Epi Info Website**, and the **Exit** button.

Note: Internet access is required to visit the Epi Info™ website.

3. The status bar at the bottom of the Epi Info™ 7 main menu screen contains important information about the application including:

    - Application status or error messages - If the application is ready to start and no errors have been encountered, **Ready** is displayed. The status bar may be hidden and can be displayed by clicking on **View > Status Bar** from the navigation menu at the top of the screen.
    - Windows language and region setting.
    - Application version number.
    - Version release date (not displayed in all versions of Epi Info™).
    - Keyboard status: **CAPS**, **NUM**, **INS**. These buttons will become active if the respective keys on the keyboard are toggled on.

Tech Support and Contact Information

The Epi Info™ Help Desk offers free technical support to all Epi Info™ users from 8 a.m. to 6 p.m. Eastern Standard Time (EST), Monday through Friday.

Epi Info Help Desk:	(404) 498-6190
Epi Info Email:	epiinfo@cdc.gov

Website

The latest version of the Epi Info™ software, tutorials, and translations can be downloaded from the Epi Info™ website (http://www.cdc.gov/epiinfo). Epi Info™ download and installation instructions are located in the appendix.

Epi Info™ Community Questions and Answers

The Epi Info™ User Community Questions and Answers website provides a resource for you to research answers to questions others have asked or to post your own questions. Join the community at the Epi Info™ community website (https://epiinfo.atlassian.net/wiki/questions).

2. Form Designer

Introduction

In Epi Info™ 7, the Form Designer and the Enter modules work together to design the data entry process and collect data. Form Designer is the tool used to design the survey, questionnaire, or form, tailor the data entry process, and specify the tab sequence. It is where you customize any data validation you want to occur when the form receives data in the Enter tool.

Data collection in Epi Info™ 7 is organized by projects. Each project can have one or many forms which can have one or many pages. On each page, one or many data entry fields are added which collect individual data elements.

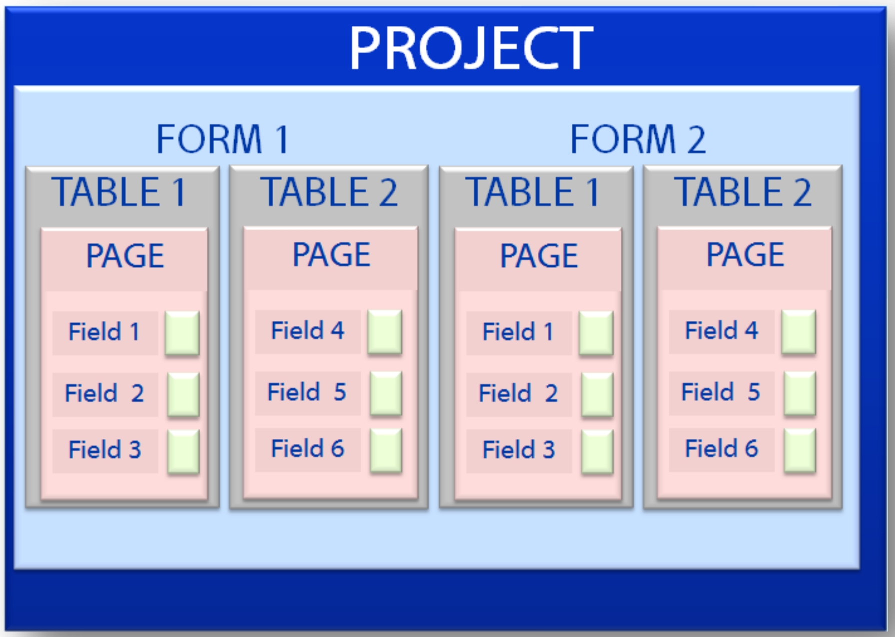


Figure 7.2: Project Organization

Fields added to a page can be any of a variety of field types corresponding to the type of data needed and the kinds of analyses that can be anticipated. Field types range from simple labels, text, numeric, and date fields to more advanced fields like drop-down lists, data grids, and command buttons. A list of the available field types is shown in Figure 2.21.

Epi Info™ 7 normally uses the Microsoft Access database format. If no changes are made to the default settings when new projects are created, data are stored in the project's MS Access data file having an ".mdb" extension. If you have access to a Microsoft SQL Server database, Epi Info™ 7 can natively use this format for data storage, instead of the MS Access format. Epi Info™ 7 handles all of the database management aspects such as the table creation, managing keys, relationships, and other information storage details. Since the process of designing the form also defines the database, the Form Designer can be regarded as a type of database design environment.

Navigate the Form Designer Workspace

To open Form Designer, click **Create Forms** from the Epi Info™ 7 - Menu, or select **Tools > Create Forms** from the toolbar.

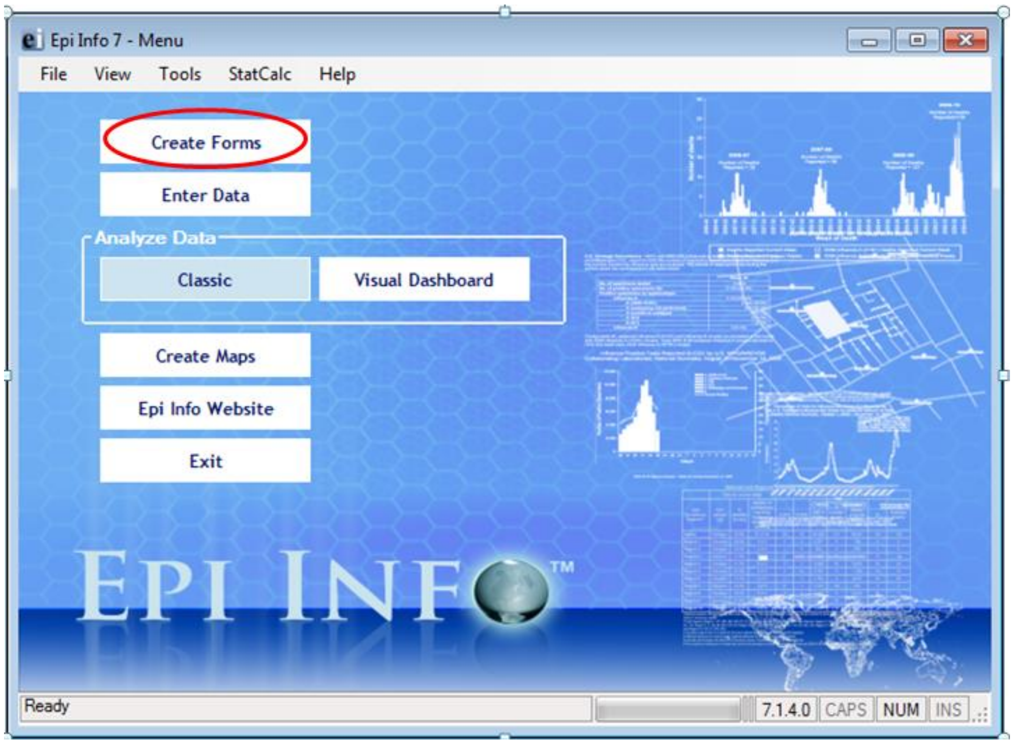


Figure 7.3: Create Forms Menu Option

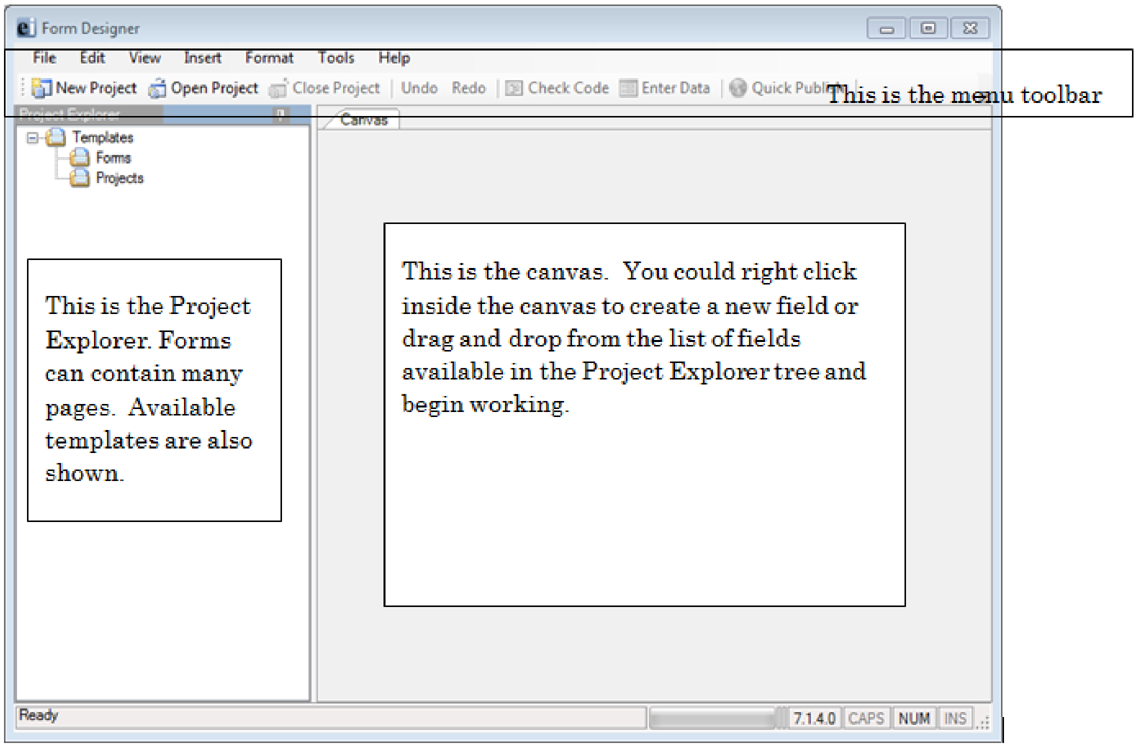

Form Designer
File Edit View Insert Format Tools Help
New Project Open Project Close Project Undo Redo Check Code Enter Data Quick Publish
This is the menu toolbar
Project Explorer
Templates
Forms
Projects
Canvas
This is the Project Explorer. Forms can contain many pages. Available templates are also shown.
This is the canvas. You could right click inside the canvas to create a new field or drag and drop from the list of fields available in the Project Explorer tree and begin working.
Ready
7.1.4.0 CAPS NUM INS

Figure 7.4 Form Designer (No Active Project)

1. The **Menu toolbar** provides an easy way to access your projects and gives you tools to edit your forms, manage your project and customize your Canvas.
2. The **Project Explorer** is located on the left of the screen and allows you to add pages, fields, and templates to a form. Each category can be collapsed or expanded by clicking their respective – or + buttons next to the category name.
3. The blank page to the right of the Project Explorer is the **Canvas** where the form is designed. Each field is defined and edited using its Field Definition dialog box. You can customize the work space by selecting fonts, colors, and grid options.

Open a Recent Project

Epi Info™ 7 is packaged with many sample projects, templates, and examples to use for helping to showcase its many features and functions. The **Sample** project is one that contains several forms already prepared with instructional data collected. Many of these forms and their associated data are used in schools of public health for teaching Epi Info™ and the science of epidemiology and statistics important to public health research and epidemiologic investigations.

To open a recent project, from the Form Designer menu, select **File > Recent Projects**.

A list of recently opened projects is shown in the menu. Even though this may be the first time opening Form Designer, the Sample project is shown in the list to get you started.

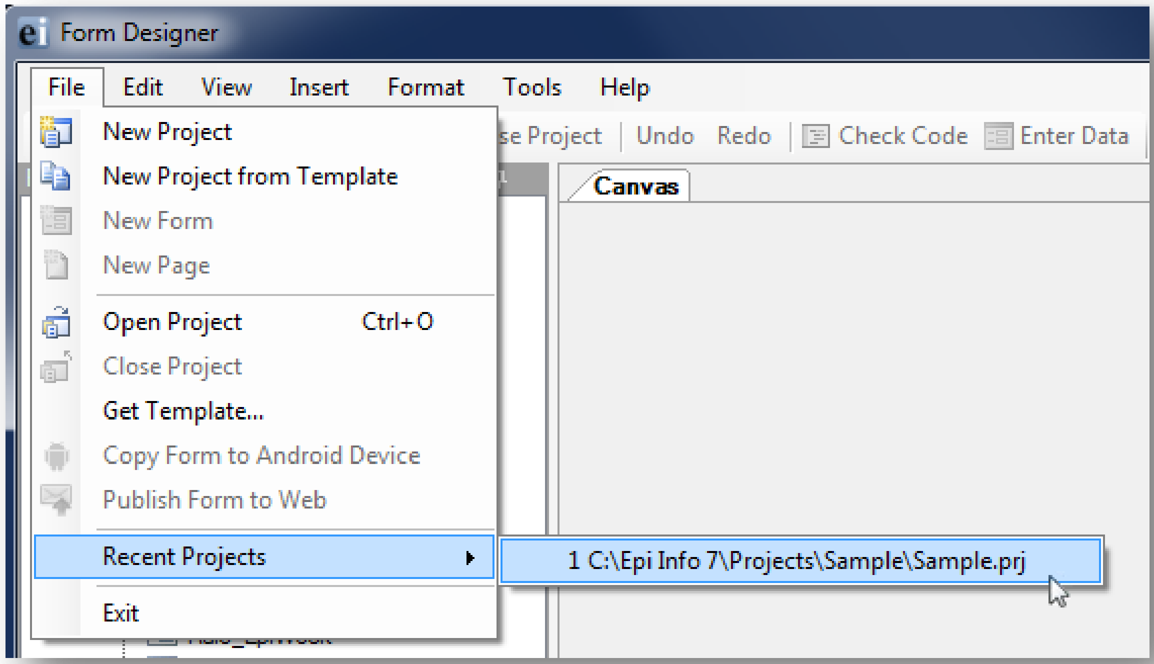


Figure 7.5: Opening a Recent Project

After selecting the Sample project, Form Designer opens to the first page of the first form – ADD Full.

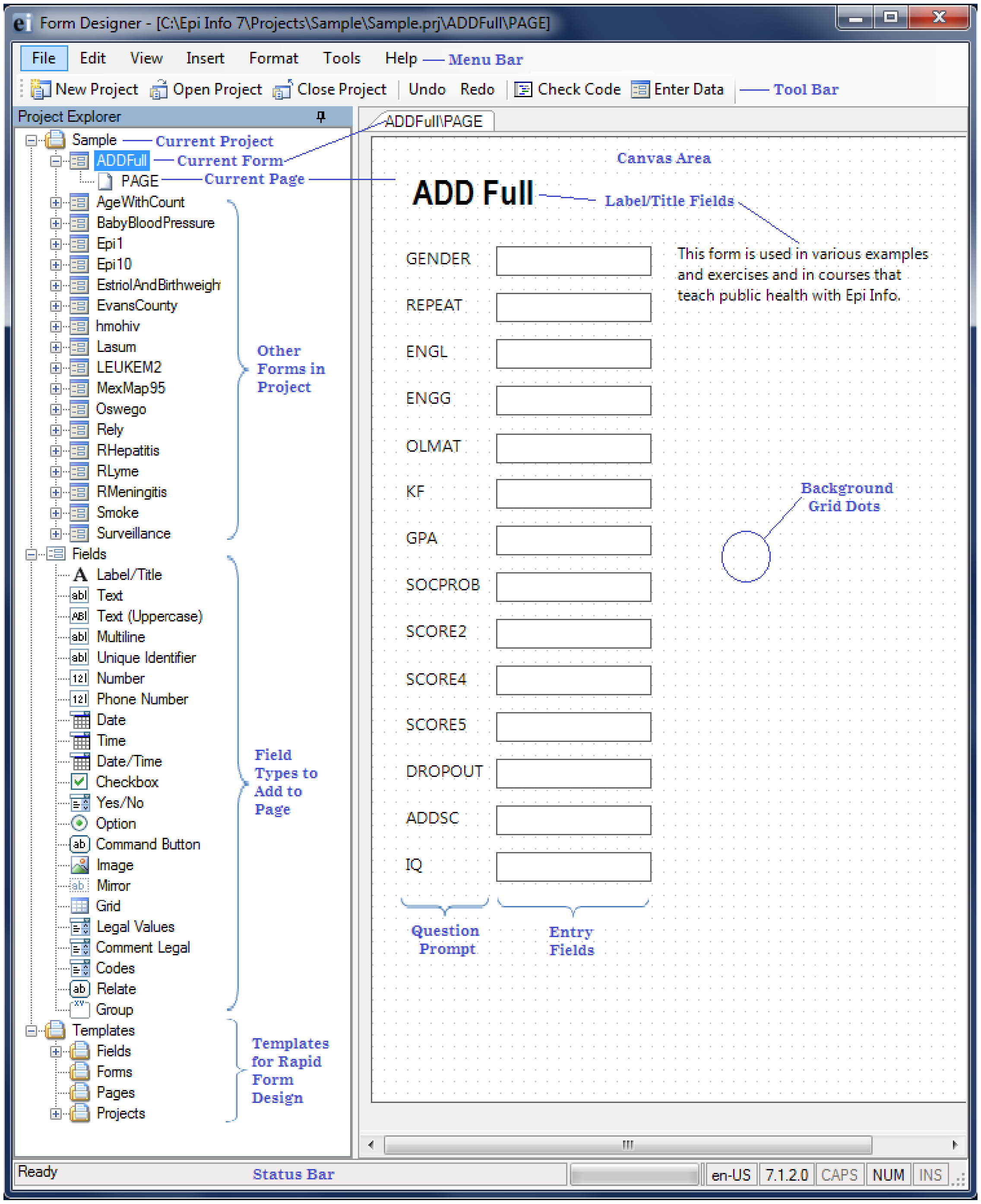


Figure 7.6: Form Designer First Page of Project

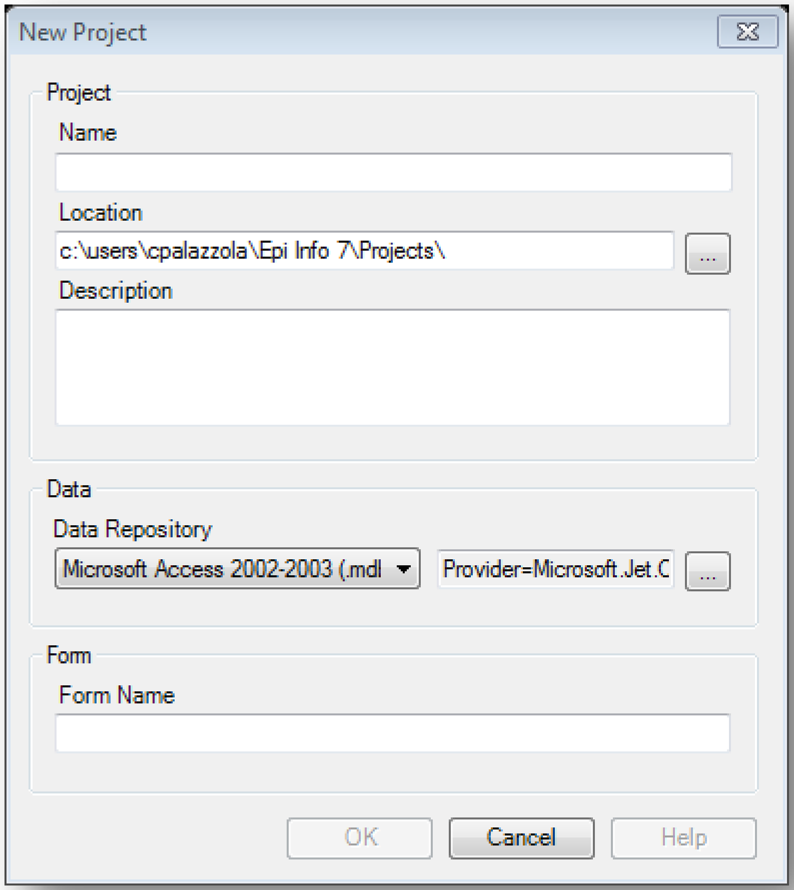
Create a New Blank Project and Form

Every Epi Info™ 7 project could contain two files; the database file and a project file (.prj) that is contained in a project folder. The project file holds the information about the location and format of the database, and whatever connection information might be required, such as a user name and password. If you choose to use the Microsoft Access format for your database, your project may also have the database file (.mdb) inside the project folder. The database file can also be saved to another location as needed. Projects that use a SQL Server database will only have the .prj file since the database is located on the server.

To create a new project and form, follow the steps below:

1. From the Epi Info main menu, select **Create Forms** or select **Tools > Create Forms** from the toolbar. The Form Designer window opens.

2. Click the **New Project** button in the toolbar or select **File > New Project**. The New Project window opens.



Location: c:\users\cpalazzola\Epi Info 7\Projects\

Data Repository: Microsoft Access 2002-2003 (.mdl Provider=Microsoft.Jet.C

Figure 7.7: New Project dialog box

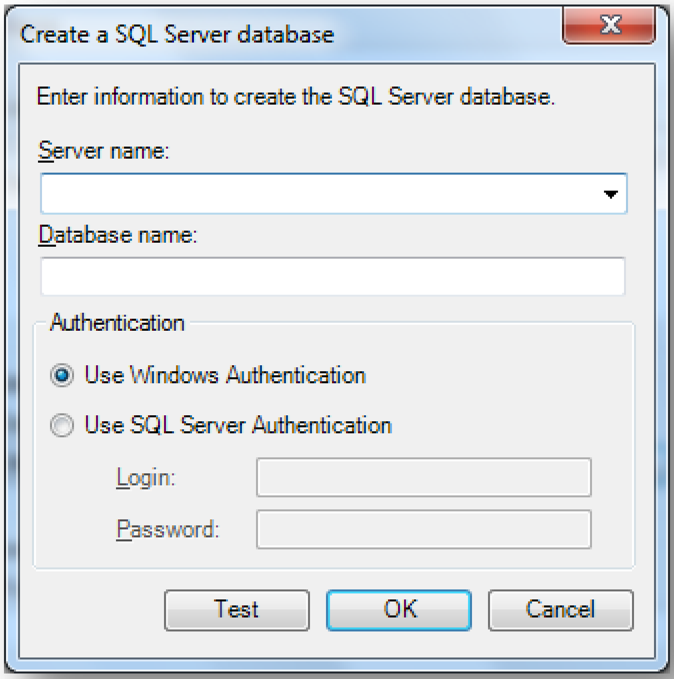
3. Enter a **Project Name**. Project names cannot contain spaces, or most non-alphanumeric characters, although underscores are permitted.

4. Set the desired project location by typing it into the Location field, or clicking the browse button. The default location for projects is the **\Epi Info 7\Projects** folder.

5. Provide a description of the project. (Optional)

6. Select the database format from the **Data Repository** drop-down list. The default option is Microsoft Access 2000-2003. However, SQL Server is also available. To use the SQL Server option, you need to have access to a SQL Server database.

7. If you selected Microsoft SQL Server for the **Data Repository**, then click the browse button to the right to enter the connection information for the SQL Server database. Contact your SQL Server database administrator for the required information requested in this dialog box.


Create a SQL Server database

Enter information to create the SQL Server database.

Server name:

Database name:

Authentication

(●) Use Windows Authentication

( ) Use SQL Server Authentication

Login:

Password:

Figure 7.8: SQL Server Database dialog box

8. Click **OK**.

9. Tab to, or select the **Form Name** field.

10. Enter a **Form Name**.

    - Use only letters, numbers, and underscores.
    - Do not start a form name with a number.
    - Do not use any spaces.

11. Click **OK**. A new blank canvas appears with the new form name and page on the tab at the top left of the canvas.

Before You Begin

Decide on a Naming Convention for Field Names

Before beginning to design your form it is helpful to have a standard way to name the data fields. While Epi Info™ will attempt to suggest a field name based on the question or prompt, in many cases these suggestions are too lengthy or cumbersome. Having a standard naming convention to assign to fields as they are added to the form will make later analyses much easier. The following suggestions may help:

- Keep field names short.
- Capitalize words within field names to help improve their readability.
  For example, "PatientLastName" is easier to read than "patientlastname".
- Use a prefix to keep associated fields together in alphabetized lists.

Customize the Form Designer

One of the most helpful things you can do *before* beginning to develop your questionnaire or survey form is to decide on the layout and overall look of the form.

Aspects to consider include the following:

- What Font do you want the Questions and Entry Fields to have? You may want to set the **Default Prompt and Input Fonts.** You specify the fonts in **Format>Set Default Prompt Font** and **Set Default Input Font.**
- How far apart should fields be spaced? You may want to adjust the settings for the grid. The grid square size and other grid settings are set in **Format > Grid Settings.**
- Will the form need to be printed? If so, what size paper will likely be used? You may want to use a corresponding page size when designing the form. You specify the page size in **Format > Page Setup.**
- What page orientation should be used, Portrait or Landscape? The answer to this may depend on what devices and platform will be used for data entry. If the form will be published to a website or uploaded to a mobile device, consider a smaller page size and an appropriate page orientation. The page orientation is set in **Format > Page Setup.**
- Would you prefer the Question or Prompt to appear above the Input Field or to the left of the Input Field? You set the question and input field positioning in **Format > Page Setup** and select **Vertical** or **Horizontal**.

- What screen resolution will likely be used for data entry? If the anticipated screen resolution is less than your usually setting, you may consider reducing your resolution while designing the form so the layout of your fields will be consistent when viewed in the Enter tool.

While you can change these format settings at any time during the form design, if these questions are considered earlier, there may be less rework required to make the form presentable than if these settings are changed after fields are added to the pages.

Set a Default Prompt or Input Field Font

Set the character font to be used as fields are added to the form using the Format menu.

Default Prompt Font

1. Format > Set Default Prompt Font. The Font dialog box appears.
2. Select a **font, font style,** and **sizes**.
3. Click **OK**.

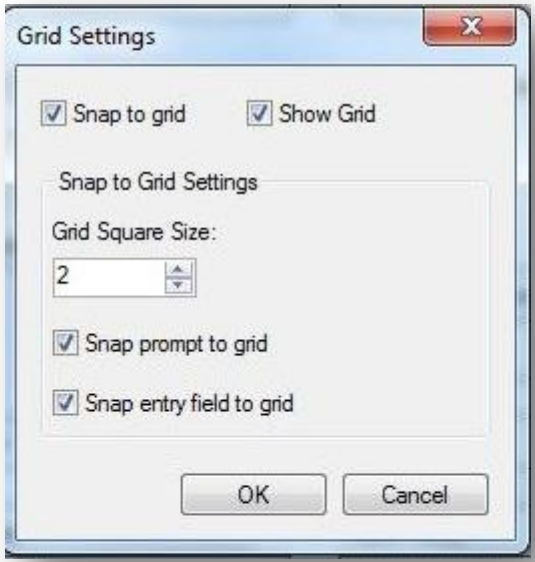
Default Input Font

1. Format > Set Default Input Font.
2. Select a **font, font style,** and **sizes**.
3. Click **OK**.

Change Grid Settings

Use the **Format** settings to customize the Form Designer canvas.

1. From the Form Designer toolbar, select **Format**. The drop-down menu opens allowing you to customize your canvas.
2. Select **Format** > **Grid Settings** to open the grid settings dialog box.



Grid Settings

[✓] Snap to grid [✓] Show Grid

Snap to Grid Settings

Grid Square Size: 2

[✓] Snap prompt to grid

[✓] Snap entry field to grid

Figure 7.9: Grid Settings

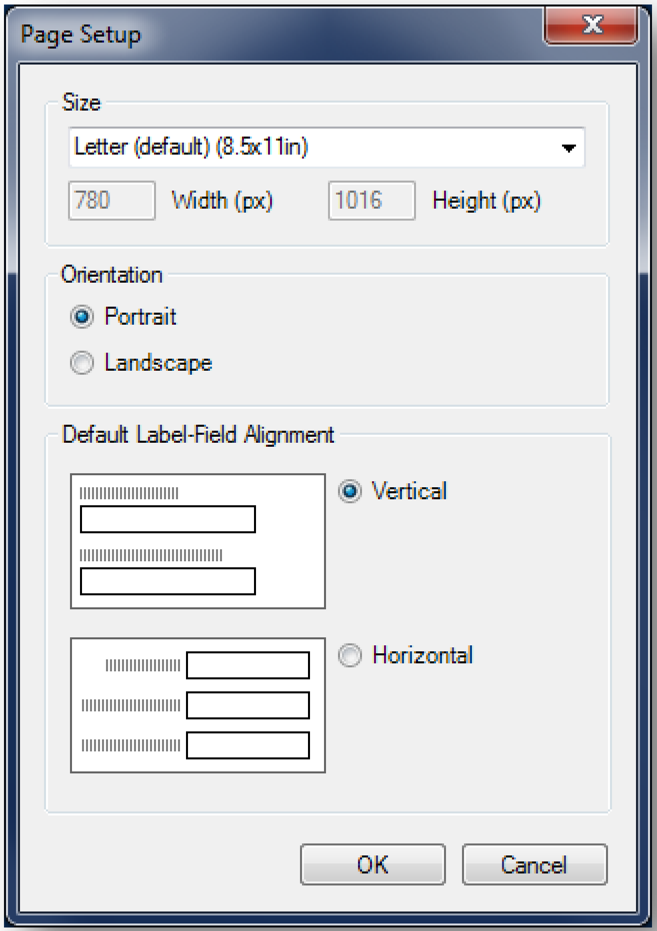
- Check the **Snap to Grid** box to force fields in the form to snap to the grid nearest the field edge.

- Check the **Show Grid** box to see the grid as the canvas background.
- Use the **up and down arrows** in the Grid Square Size field to alter the displayed width between grid lines.
- Select the **Snap prompt to grid** or **Snap entry field to grid** check boxes depending on the snapping effect you want to use.

3. Click **OK**. The Form Designer page appears with new settings.

Set the Page Size, Orientation, and Default Prompt Alignment

The Page Size, Orientation, and Default Prompt Alignment are set using the Page Setup dialog.



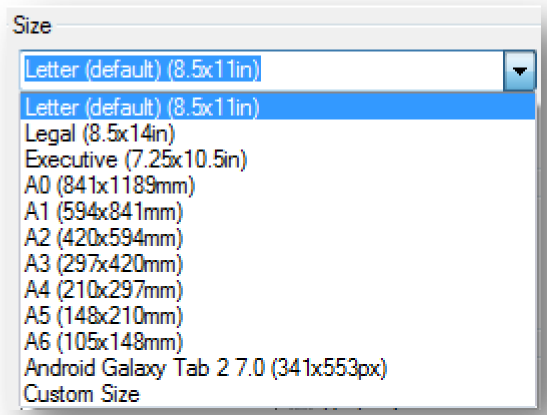
Letter (default) (8.5x11in)
780 Width (px) 1016 Height (px)

[✓ Portrait] Landscape

[✓ Vertical] Horizontal

Figure 7.10: Page Setup

1. On the Form Designer menu, select **Format > Page Setup**.
2. Set the page size to be any of the pre-configured sizes listed in the Size drop-down list or use a Custom Size and specify the Width and Height in pixels.


Size

Letter (default) (8.5x11in)

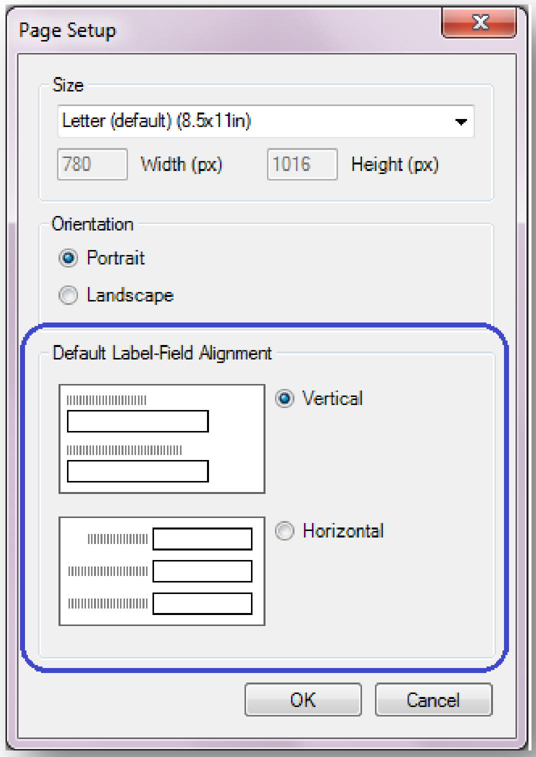
Figure 7.11: Page Size Setup

3. Set the preferred page orientation – Portrait or Landscape.

    - Portrait orientation is good for data entry forms that need to be printed and when the data entry is done from printed forms that have a portrait orientation. Portrait orientation may also be preferred for forms uploaded to small mobile devices such as smart phones or tablets.

    - Landscape orientation is better for forms intended to be published to a website for entry using an Internet browser on a desktop or laptop computer.

    Set the default alignment to be vertical so the field is below the prompt and left aligned, or horizontal where the field is to the right of the prompt on the same row.


Page Setup

Size: Letter (default) (8.5x11in)

780 Width (px) 1016 Height (px)

Orientation: [✓ Portrait] Landscape

Default Label-Field Alignment: [✓ Vertical] Horizontal

Figure 7.12: Page Setup dialog showing Default Label-Field Alignment

Insert a Background Image or Color

1. From the Form Designer toolbar, select **Format > Background**. The Background dialog box opens.

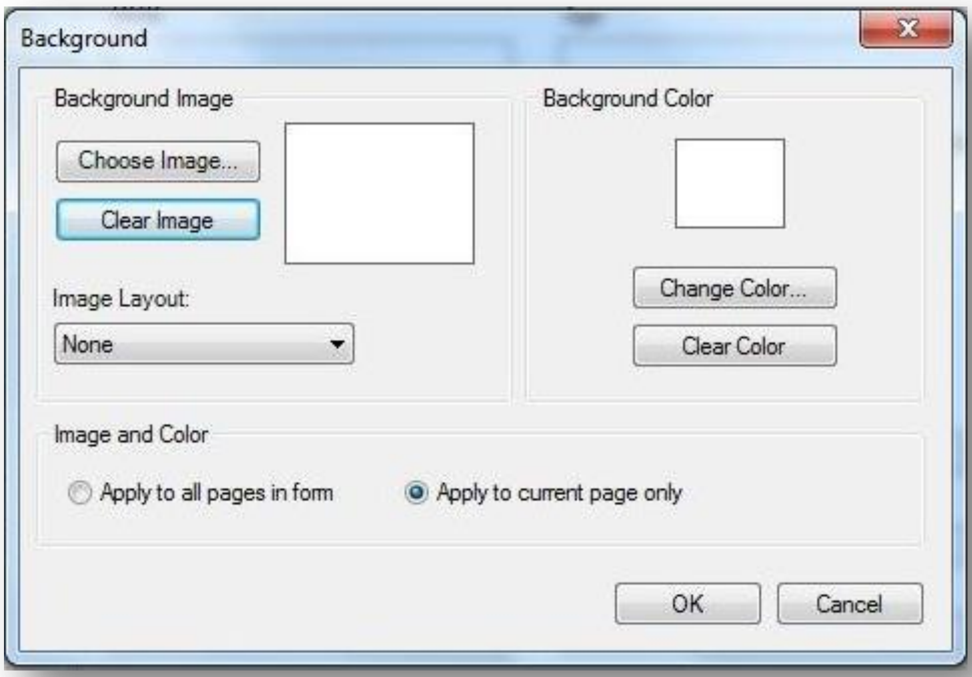


Figure 7.13: Background Format

2. To set a background image, use the Background Image section of the dialog, click **Choose Image**. The Background Image box opens.

3. Locate the image file. Click **Open**. The selected image appears in the Background Image dialog box. Image formats include bitmap (.bmp), picture (.ico), and JPEG (.jpg).

4. Use the **Image Layout** drop-down list to customize the image on the screen (**None, Tile, Center, and Stretch**).

5. From the Image and Color section, use the option buttons to **Apply to all pages** or **Apply to the current page only**.

6. Click **OK**. The image appears in the form.

7. To remove the image, select **Clear Image** from the Background Image box.

8. To set a background color, click **Change Color**. The Color dialog box opens.

9. Select a **background color** from the palette or select **Define Custom Colors** to enter a more specific color request.

10. Click **OK**. The selected color previews in the background box.

11. From the Image and Color section, use the option buttons to **Apply to all pages** or **Apply to the current page only**.

12. Click **OK**. The color appears in the form as a background.
13. To remove the color, select **Clear Color** from the Background Color box.

Create a New Project from a Project Template

Another way to create a new project is to use a pre-defined project template. This option creates the project with all of the forms, pages, fields, and Check Code that are specified in the project template. Epi Info™ 7 comes with a number of sample project templates to demonstrate its many features. These templates are located in the Project Explorer under **Templates > Projects**.

To use an existing project template, follow the steps below:

1. Select **File > New Project from Template**. The New Project from Template window opens listing the available project templates.
2. Select the desired project template.
3. Confirm the name and location of the new project, and the data format as shown in the window and make changes as needed.
4. Click **OK.** The Form Designer will begin constructing the forms and fields and after a few moments, the first page of the project opens in the canvas.

Create a New Form in an Existing Project

1. Select **File > New Form** or right click on the project folder in the Project Explorer and select **Add Form**. The form dialog box opens.
2. Type a **Form Name**.
3. Click **OK**. The new form appears in the Project Explorer, and the first page appears in the canvas area.

Open an Existing Project

There are two ways to open an existing project.

If the project you want to open is one of the four most recent projects opened, select **File > Recent Projects** and select the project from the list.

Otherwise, follow these steps:

Click the **Open Project** button from the toolbar or select **File > Open Project**. The Open dialog appears

1. Select a **project file (.prj)** from the Epi Info™ Projects folder.

Note: Epi Info™ will only display files in the Epi Info Project File format (*.prj).

2. Click **Open**. The project appears on the canvas.

Close Project

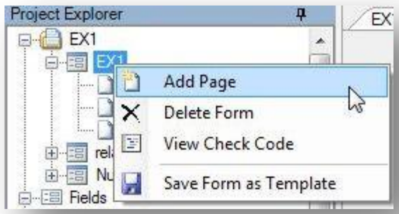
1. Click the **Close Project** button from the toolbar or select **File > Close Project**.
2. The project closes.

Note: Epi Info™ automatically saves all changes to the project. There is no save function in Form Designer.

Add or Insert a Page

Add a Page

The Add Page function will add a blank page at the end of the form. To add a page, right click on the form in the Project Explorer and select **Add Page** from context menu or select **Insert** > **Page** > **Add Page** from the toolbar.



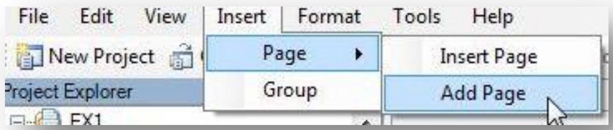


Figure 7.14: Add a Page

Enter a page name in the **Set Page Name** dialog box and click **OK**. The blank page appears on the canvas and in the Project Explorer at the end of the form.

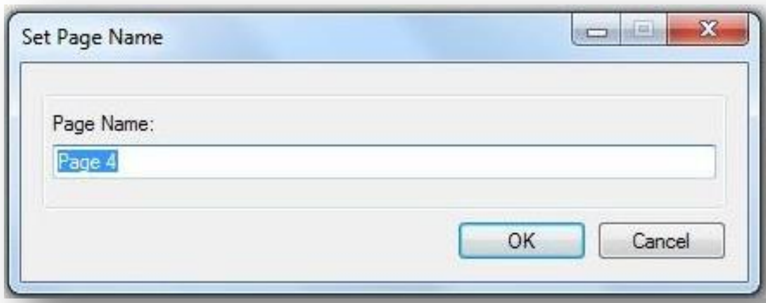
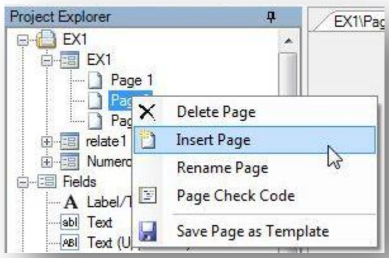


Figure 7.15: Set Page Name

Insert Page

The Insert Page function will add a blank page immediately before the current page displayed on the canvas. To insert a page, right click on the desired page in the Project Explorer and select Insert Page from the context menu or select **Insert** > **Page** > **Insert Page** from the toolbar



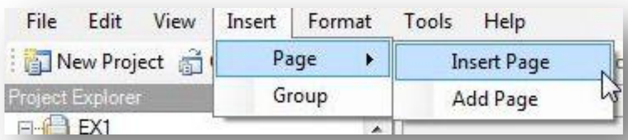


Figure 7.16: Insert Page

The new page appears on the canvas and in the Project Explorer immediately before the current page.

Name a Page

1. Right click on a page in the Project Explorer. Select **Rename Page** from the context menu.

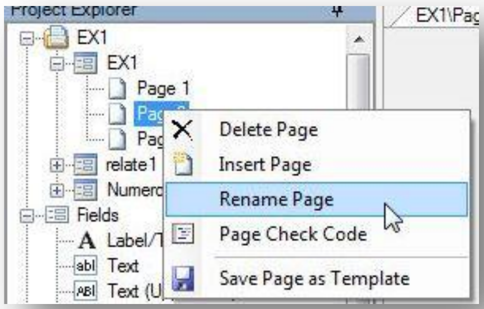


Figure 7.17: Rename Page

2. The **Page Name** dialog box opens. Enter a name in the Page Name field.

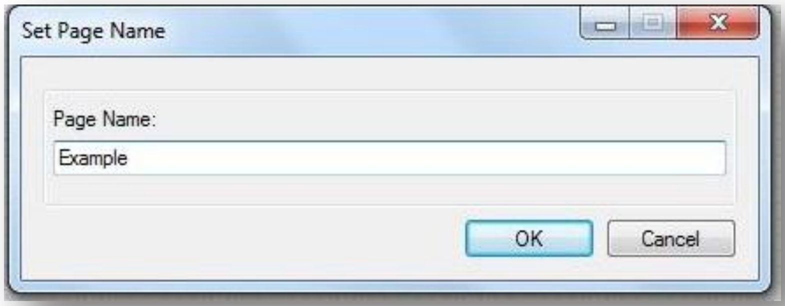


Figure 7.18: Set Page Name

3. Click **OK**. The page name appears in the list of pages.

Delete a Page

From the Project Explorer, right click on the page you want to delete and select **Delete Page** or select **Edit > Delete Page** from the toolbar.

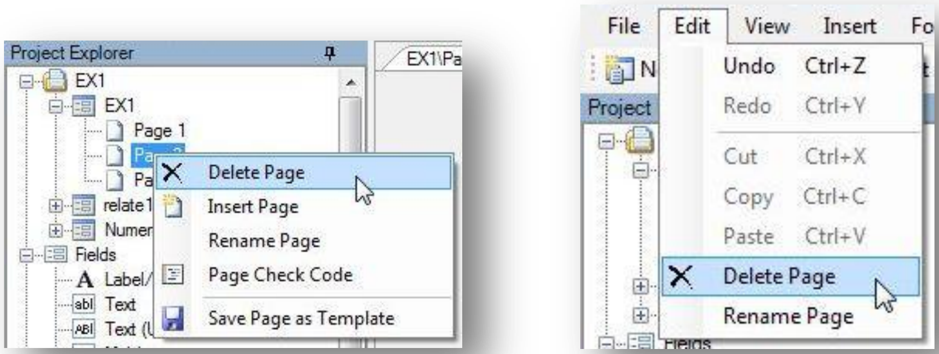
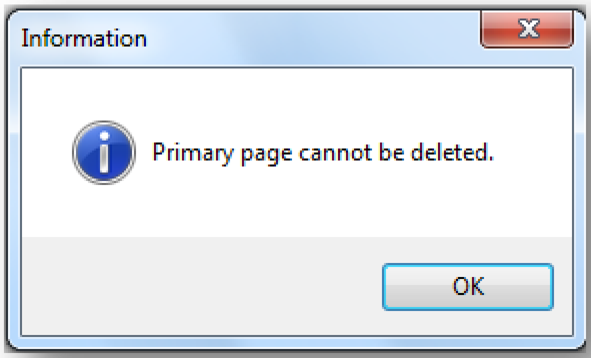


Figure 7.19: Delete Page

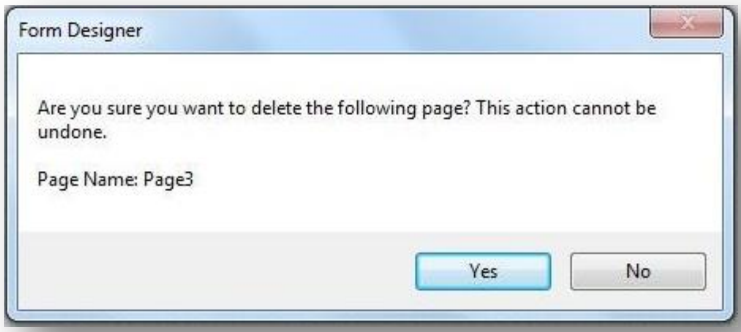
Note: If there is only one page to the form, then you cannot delete the only page.



Primary page cannot be deleted.

Figure 7.20: Primary Page cannot be Deleted

Otherwise, a dialog box opens prompting you to confirm deletion. Click **Yes**. The page is deleted.



Are you sure you want to delete the following page? This action cannot be undone.

Page Name: Page3

Figure 7.21: Delete Page Confirmation dialog box

Undo / Redo

Most actions performed on the canvas, such as moving fields, copying and pasting fields, and changing field alignment, are recorded in the memory of the tool. This allows you to undo any of those actions.

Click **Undo** from the toolbar or **Ctrl + Z** to undo the most recent action in the list. Form Designer begins recording actions when it is opened. Therefore, you can undo all actions from the most recent to the first action performed since opening Form Designer.

Click **Redo** from the toolbar or **Ctrl + Y** to redo the most recent action undone. As with Undo, the Form Designer remembers the actions retracted by clicking Undo. Therefore you can Click Redo repeatedly to reinstate the actions retracted by Undo, up to the first time Undo was clicked.

Form Designer only remembers actions in the Redo list that were undone by clicking Undo. As soon as you make any other changes directly to the canvas, other than clicking Undo, then the Redo list is discarded. ***Redo can only reinstate actions that were undone***—it cannot be used to repeat actions made directly to the canvas.

Check Code

Epi Info™ allows the creator of the questionnaire or survey to guide the data entry process to produce range checking, automatic skip patterns and coding of variables. This is accomplished using Check Code. Many other more complex functions are possible such as mathematical or logical operations, checking the value of a field against the value of one or more other fields, and displaying helpful dialogs or custom error messages. Click the **Check Code** button to open the Check Code Editor. For information about the Check Code Editor, refer to the Check Code section of the user guide.

Enter Data

The Enter tool is used to enter data into the data entry forms in Epi Info™ 7. Click the **Enter Data** button to open the current form in the Enter tool. Only independent, stand-alone forms or Parent level forms can be opened in Enter. If the form is a Related Form—as in a child form related to another parent form—then you cannot open the Related Form by clicking the Enter Data button. You must open the top-most parent form first, and then use the parent form's relate button to access the child form. This is required in order to keep the record hierarchy according to the related table relationship.

If a form does not yet contain a data table, Epi Info™ will prompt you to create a data table in this step. For information regarding the Enter tool, refer to the Enter Data section of the user guide.

Fields

A Field is sometimes called a data entry field or input field. The field is usually where the answers to the questions are entered, although Epi Info™ 7 has a few field types that don't receive any data such as Label Fields and Command Buttons.

Fields usually contain a question or prompt or some text describing the data that will be collected. In Epi Info™ 7, different fields are provided to collect different types of information. A list of available fields is shown in the Project Explorer under the Fields category or when you right click on the Canvas and select **New Field**.

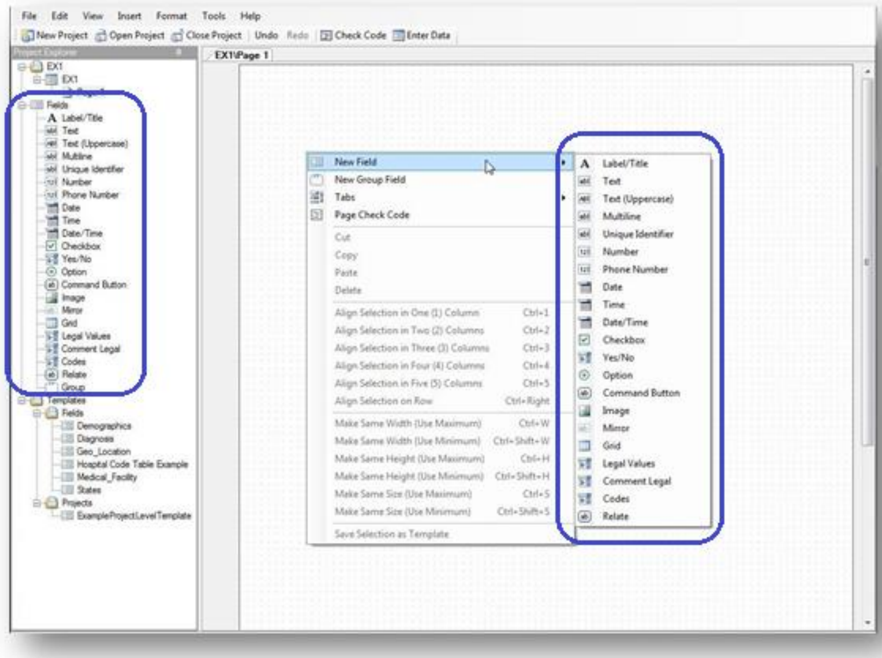


Figure 7.22: New Field context menu

Field Attributes

Fields have properties or attributes that help control how or if they receive data. Each field type has a set of available field attributes; however, some options may not be shown or may be disabled (grayed out) depending on the field type selected.

- **Required** field attribute makes a field mandatory in data entry. If a page contains fields marked with the Required attribute, the Enter tool will not allow record navigation or the saving of a partially completed record until a value is supplied in all Required fields.

Use With Caution! The Required attribute can potentially cause gridlock if, for any reason, the required information is not available at the time of data entry. It is best to use this attribute sparingly to avoid this situation.

Additionally, Check Code can be used to set or unset a field's required status. For more information on this feature, refer to the topics in the Check Code Command Reference on the SET-REQUIRED and SET-NOT-REQUIRED commands.

- **Read Only** – prevents data entry into the field. This attribute is used for fields that hold calculated or prepopulated values that should not be edited. The cursor cannot be placed in the field and therefore no data entry can occur. Also, since the cursor cannot enter a field with this attribute, Check Code written for this field will not have a chance to run.

The Read Only attribute is useful for calculated fields that will not be changed directly. Read Only cannot be used in combination with the Required attribute because those attributes are mutually exclusive.

- **Retain Image Size** – used only for image fields, this attribute maintains the size of the original image and does not alter the size to fit the image box in the form.
- **Range** – enables a valid range with lower and upper limit fields. Only Number and Date field types can have range values. If a value is entered that is less than the lower limit or greater than the upper limit, you will get a warning message. Missing values are accepted unless the field's Required attribute is checked.
- **Repeat Last** – causes the field to be automatically populated with the last value displayed for that field. This may speed up data entry for fields that often have a value the same as the last record entered. For example, if entering many records from a given clinic, then a "Clinic Name" field could have the Repeat Last attribute checked so it will be automatically filled in from the clinic name used in the last record.
- **Pattern** – the format that a response must be entered in the field.
- **Prompt Font --** the font for the text entered in the Question or Prompt text box.
- **Field Font --** the font for the data entered into the field.

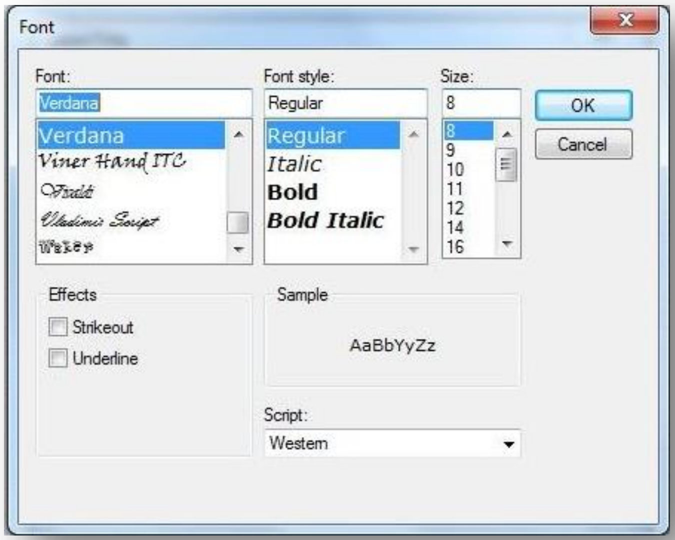


Figure 7.23: Font dialog box

Field Type	Required	Read Only	Retain Image Size	Range	Repeat Last	Pattern	Web Survey Compatible	Companion for Mobile Compatible
Label/Title							X	X
Text	X	X			X		X	X
Test (Uppercase)	X	X			X			X
Multiline	X	X			X		X	X
Unique Identifier								
Number	X	X		X	X	X	X	X
Phone Number	X	X			X	X		
Date	X	X		X	X		X	X
Time	X	X			X		X	X
Date-Time	X	X			X			
Checkbox		X			X		X	X
Yes-No	X	X			X		X	X
Option							X	X
Command Button								X
Image			X					X
Mirror								
Grids								
Legal Values	X	X			X		X	X
Comment Legal	X	X			X		X	X
Codes	X	X			X			
Relate								
Group							X	X

Figure 7.24: Field Attributes

Add a New Field

Click and drag the type of field you want to add from the Project Explorer to the approximate location on the canvas.

Alternatively, right click on the canvas in the approximate location you want to add a field and select the field type from the **New Field** context menu.

A **Field Definition** dialog box appears. The contents of the Field Definition dialog depend on the type of field being added.

Label/Title

The **Label/Title** field allows you to have a title on the form or to give instructions or other information. Since this field type does not allow any data entry, it does not have Check Code, and is not searchable. The following figure is an example of a Label/Title field in Enter (circled in blue) based on the FoodHistory form of the sample EColi project.

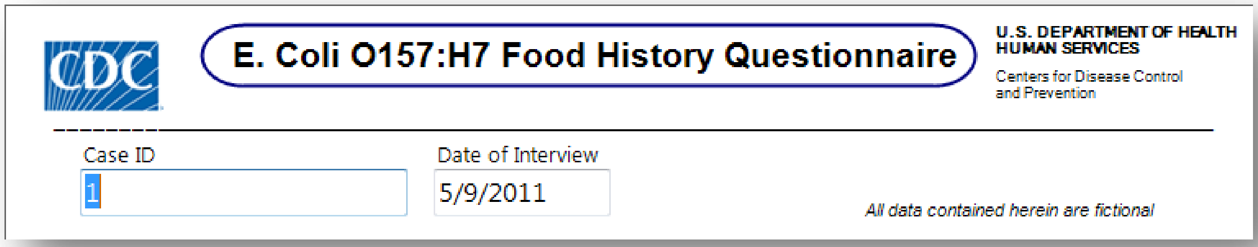

E. Coli O157:H7 Food History Questionnaire
U.S. DEPARTMENT OF HEALTH HUMAN SERVICES
Centers for Disease Control and Prevention
Case ID
1
Date of Interview
5/9/2011
All data contained herein are fictional

Figure 7.25: Label/Title Field

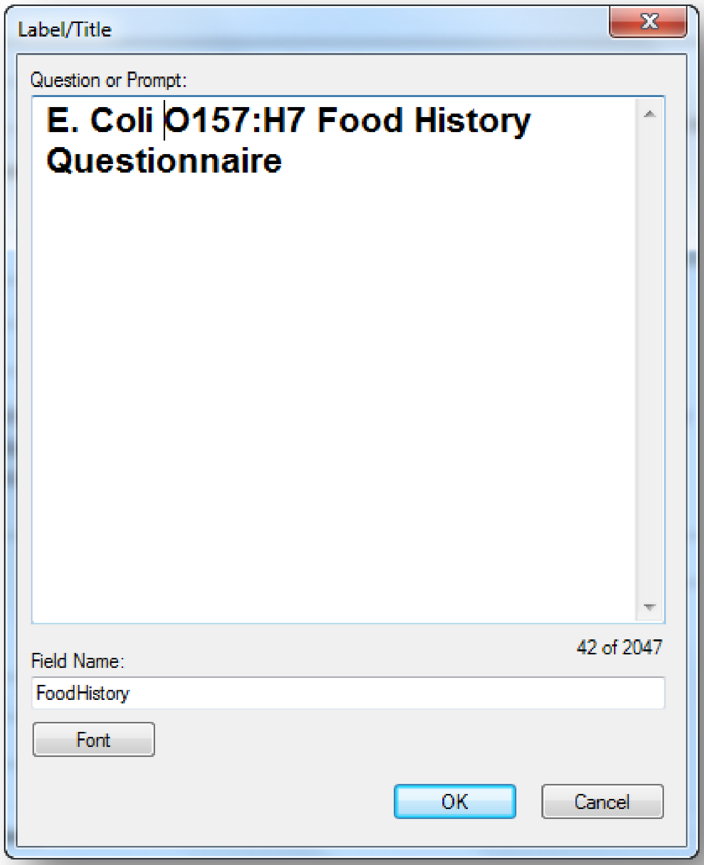
To complete the Label/Title Field Definition dialog:

1. Enter the **Question** or **Prompt**. The text entered in this field will display on the canvas.

2. Click in the **Field Name** text box. Epi Info automatically suggests a field name based on the Question or Prompt, however, it is very important that field names be short, intuitive, and usable. The field name is used for data validation in Check Code and when doing analyses. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the field name at this time. Field names cannot be changed after data collection starts.

3. Click the **Font** button to adjust the Question or Prompt font settings. ***Titles typically are configured with larger font or bolder text***.



E. Coli O157:H7 Food History Questionnaire

42 of 2047

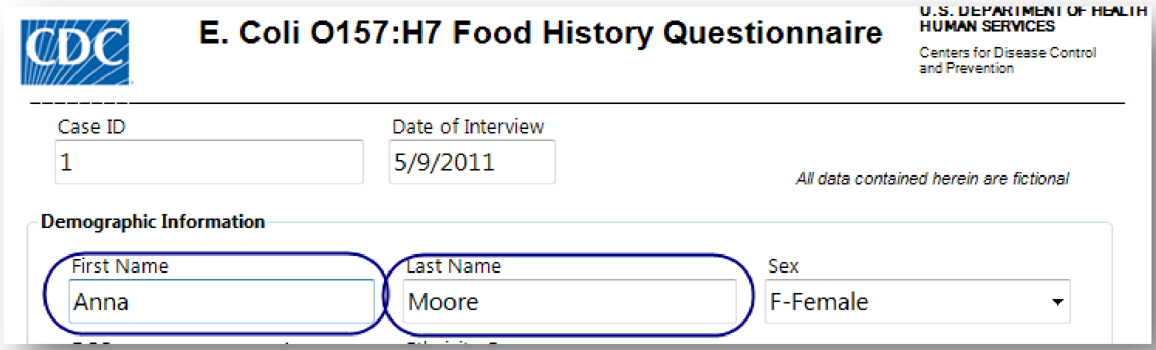
FoodHistory

Figure 7.26: Label/Title Field Definition dialog box

4. Click **OK**. The Label/Title appears on the canvas.

Text Field

The **Text** field is one of the most generic and common data entry fields used to capture text type data—letters, numbers, and symbols. Text fields hold up to 255 characters in a single line. You can restrict the number of characters entered by specifying a maximum field size. Text fields are often used for short questions such as name, address, occupation. The following figure provides an example of how a Text field appears in Enter (circled in blue) based on the FoodHistory form of the EColi project.


E. Coli O157:H7 Food History Questionnaire
U.S. DEPARTMENT OF HEALTH HUMAN SERVICES
Centers for Disease Control and Prevention
All data contained herein are fictional
Demographic Information

Case ID: 1
Date of Interview: 5/9/2011
First Name: Anna
Last Name: Moore
Sex: F-Female

Figure 7.27: Text field

To complete the Text Field Definition dialog box:

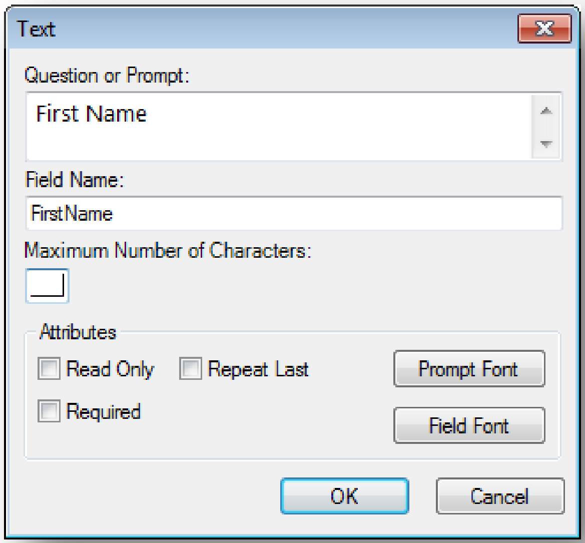
1. Enter the **Question** or **Prompt**. The text entered in this field will display on the canvas and prompt the user to enter a response.

2. Click in the **Field Name** text box or press the tab key. Epi Info automatically suggests a field name based on the Question or Prompt, however, it is very important that field names be short, intuitive, and usable. The field name is used for data validation in Check Code and when doing analyses. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the field name at this time. Field names cannot be changed after data collection starts.

3. Optionally, you can limit how long the value can be typed into this field by specifying the maximum number of characters using a number (e.g. 10 for ten characters). If the maximum number of characters box is left blank, text fields can accept up to 255 characters.

   Note: Entering a maximum number of characters is optional.



First Name

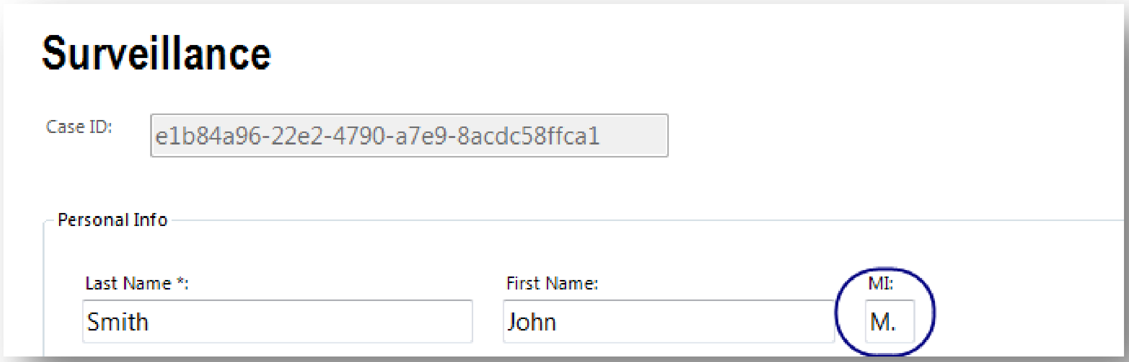
FirstName

Figure 7.28: Text Field Definition dialog box

4. Click **OK**. The Text field appears on the canvas.

Text Uppercase

The **Text Uppercase** field is used to help standardize the data entered. All letters typed in this field are automatically in ALL CAPS. This field draws emphasis to the information collected on the form. This field type is not supported on web surveys. The following figure provides an example of how a Text Uppercase field appears in Enter (circled in blue) based on the Surveillance form of the Sample project.



Case ID: e1b84a96-22e2-4790-a7e9-8acdc58ffca1

Last Name *: Smith
First Name: John
MI: M.

Figure 7.29: Text (Uppercase) field

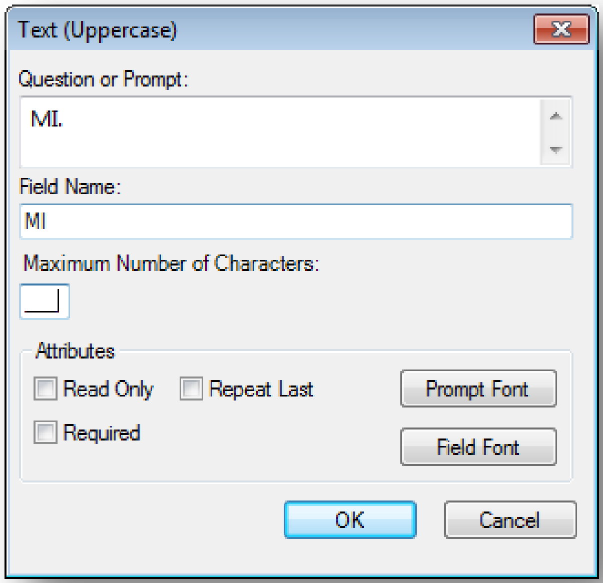
To complete the Text Uppercase Field Definition dialog box:

1. Enter the **Question** or **Prompt**. The text entered in this field will display on the canvas and prompt the user to enter a response.

2. Click in the **Field Name** text box or press the tab key. Epi Info automatically suggests a field name based on the Question or Prompt, however, it is very important that field names be short, intuitive, and usable. The field name is used for data validation in Check Code and when doing analyses. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the field name at this time. Field names cannot be changed after data collection starts.

3. Optionally, you can limit how long the value can be typed into this field by specifying the maximum number of characters using a number (e.g. 10 for ten characters). If the maximum number of characters box is left blank, text fields can accept up to 255 characters.



Question or Prompt: MI.

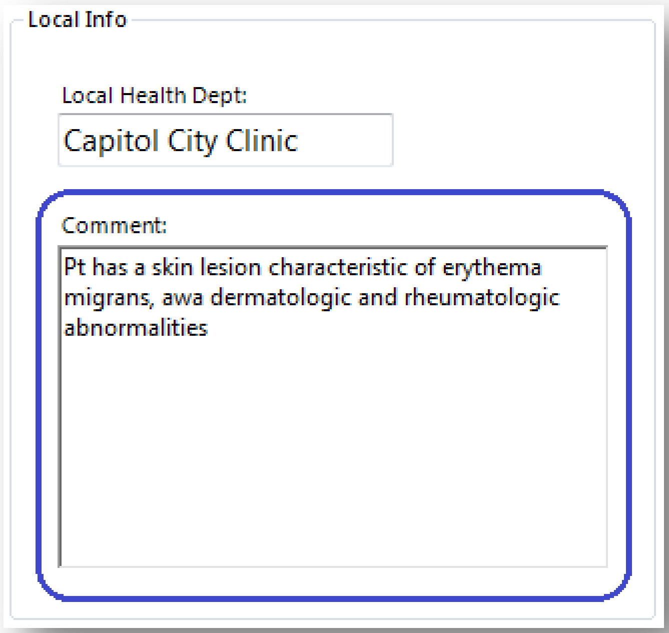
Field Name: MI

Figure 7.30: Text (Uppercase) Field Definition dialog box

4. Click **OK**. The Text Uppercase field appears on the canvas.

Multiline

The **Multiline** field is similar to the Text field but allows for several lines of text. This field type allows you to enter a large amount of text based on the question or prompt. The benefit of using this type of field is that a Multiline field can hold up to 1 gigabyte of data or approximately two million characters. However, you can’t specify the maximum number of characters that can be entered into a Multiline field the way you can with a Text field. Also, some analyses are more difficult with Multiline data. For example, you can't sort data based on values in a Multiline field, so if sorting is important for your work, may be better to use another type of field. Multiline fields are most useful when capturing lengthy narratives and wordy, free-form, and descriptive data. The following figure provides an example of how a Multiline field appears in Enter (circled in blue) based on the Surveillance form of the Sample project.



Local Info

Local Health Dept:

Capitol City Clinic

Comment:

Pt has a skin lesion characteristic of erythema migrans, awa dermatologic and rheumatologic abnormalities

Figure 7.31: Multiline field

To add a Multiline field:

1. Open the **Multiline Field Definition** dialog box.
2. Enter the **Question** or **Prompt**. The text entered in this field will display on the canvas and prompt the user to enter a response.

3. Click in the **Field Name** text box or press the tab key. Epi Info automatically suggests a field name based on the Question or Prompt, however, it is very important that field names be short, intuitive, and usable. The field name is used for data validation in Check Code and when doing analyses. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the field name at this time. Field names cannot be changed after data collection starts.

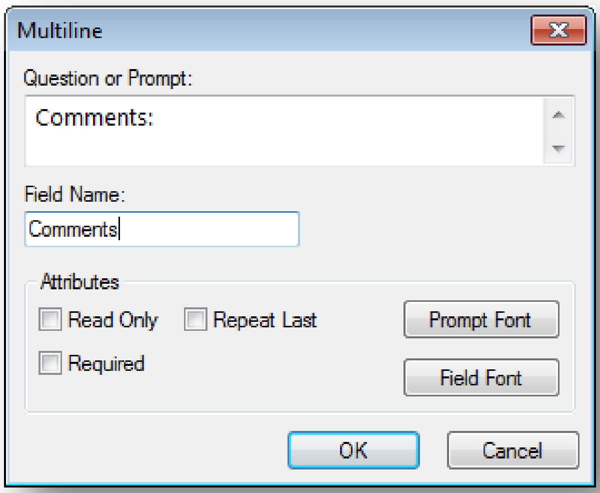


Figure 7.32: Multiline Field Definition dialog box

4. Click **OK**. The Multiline field appears on the canvas.

Number

The **Number** field only accepts numeric values. Letters and most symbols cannot be entered into this type of field, however the negative sign (-) and the decimal point (.) are allowed. Number fields can have a valid upper and lower range. For numbers that must have a specific number of decimals, you can specify a required pattern. For example, if your data requires two decimal places, your number field could have the pattern "##.##" which would allow a number between -9.99 to 99.99.

Note: The negative sign takes one of the pattern spaces.

Therefore, the number "-7" entered into a field with this pattern would appear as "-7.00", whereas the number "5" would appear as "05.00". The leading zeros are not included in calculations, analyses, or output. The following figure provides an example of how a Number field appears in Enter (circled in blue) based on the FoodHistory form of the EColi project.

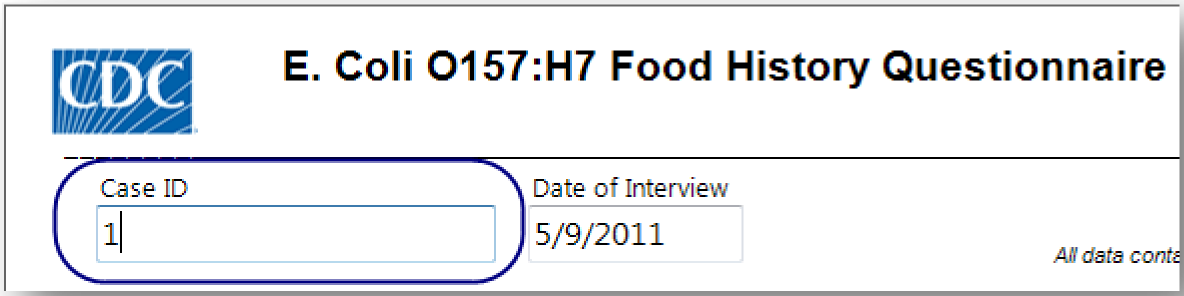


Figure 7.33: Number field

To add a Number field:

1. Open the **Number Field Definition** dialog box.

2. Enter the **Question** or **Prompt**. The text entered in this field will display on the canvas and prompt the user to enter a response.

3. Click in the **Field Name** text box or press the **tab** key. Epi Info automatically suggests a field name based on the Question or Prompt, however, it is very important that field names be short, intuitive, and usable. The field name is used for data validation in Check Code and when doing analyses. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the field name at this time. Field names cannot be changed after data collection starts.

4. Optionally, select a pattern from the **Pattern** drop-down list (optional). You can also create a new pattern by typing in the desired pattern using the # symbol and the decimal point as needed. The default setting is None.

5. To limit the range of responses, check the **Range** checkbox (optional). Type a lower and upper limit in the **Lower** and **Upper** fields.

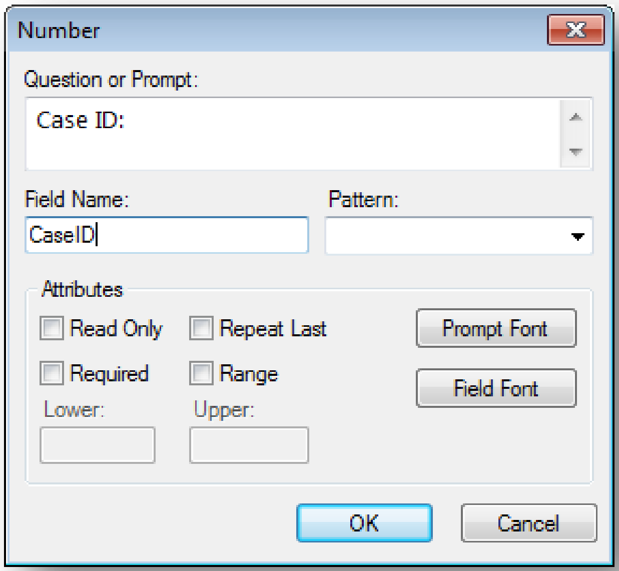
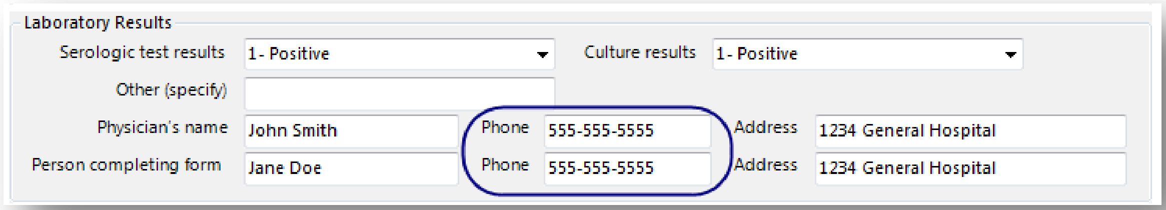


Figure 7.34: Number Field Definition dialog box

6. Click **OK**. The Number field appears on the canvas.

Phone Number

The **Phone Number** field is similar to a number field, but with a pre-defined list of common phone number patterns. As with the Number field, letters and most symbols are not allowed. This field type is not supported on Web Survey. The following figure provides an example of how a Phone Number field appears in Enter (circled in blue) based on the Case Report form of the Lyme project.


Laboratory Results

Serologic test results	1- Positive	Culture results	1- Positive		
Other (specify)					
Physician's name	John Smith	Phone	555-555-5555	Address	1234 General Hospital
Person completing form	Jane Doe	Phone	555-555-5555	Address	1234 General Hospital

Figure 7.35: Phone Number field

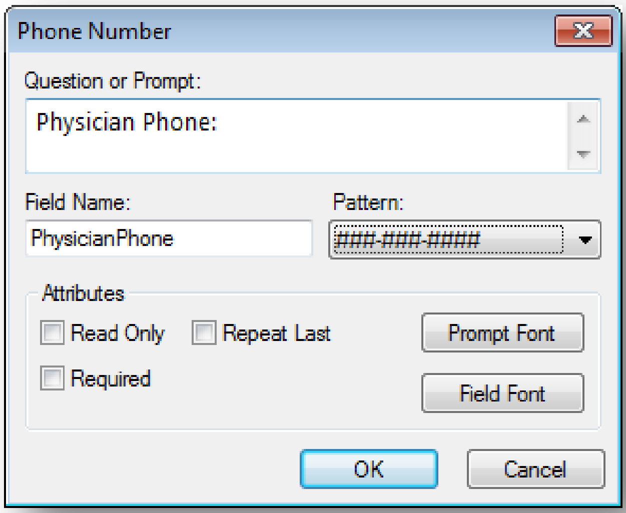
To add a Phone Number field:

1. Open **the Phone Number Field** Definition dialog box.
2. Enter the **Question** or **Prompt**. The text entered in this field will display on the canvas and prompt the user to enter a response.

3. Click in the **Field Name** text box or press the tab key. Epi Info automatically suggests a field name based on the Question or Prompt, however, it is very important that field names be short, intuitive, and usable. The field name is used for data validation in Check Code and when doing analyses. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the field name at this time. Field names cannot be changed after data collection starts.

4. Optionally, select a pattern from the **Pattern** drop-down list. The default setting is None.


Phone Number

Question or Prompt: Physician Phone:

Field Name: PhysicianPhone

Pattern: ###-###-####

Attributes

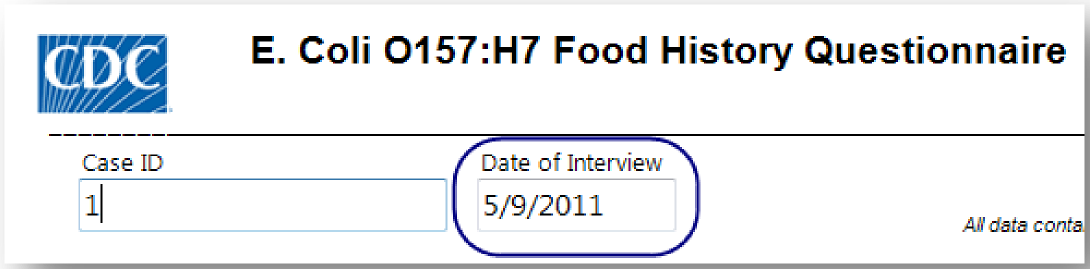
Read Only [ ] Repeat Last [ ] Required [ ]

Figure 7.36: Phone Number Field Definition dialog box

5. Click **OK**. The Phone Number field appears on the canvas.

Date

The **Date** field is used to enter a date. The date pattern for data entry is determined by the computer's regional settings where the pattern can be modified. The date pattern cannot be changed within Epi Info™ 7. The following figure provides an example of how Date fields appear in Enter (circled in blue) based on the FoodHistory form of the EColi project.


E. Coli O157:H7 Food History Questionnaire

Case ID: 1

Date of Interview: 5/9/2011

All data conta

Figure 7.37: Date field

To add a Date field:

1. Open the **Date Field Definition** dialog box.
2. Enter the **Question** or **Prompt**. The text entered in this field will display on the canvas and prompt the user to enter a response.
3. Click in the **Field Name** text box or press the tab key. Epi Info automatically suggests a field name based on the Question or Prompt, however, it is very important that field names be short, intuitive, and usable. The field name is used for data validation in Check Code and when doing analyses. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the field name at this time. Field names cannot be changed after data collection starts.

4. To limit the range of responses, check the **Range** checkbox (optional). Enter a lower and upper date range into the Lower and Upper fields or select a date using the date calendar by clicking on the calendar icon.

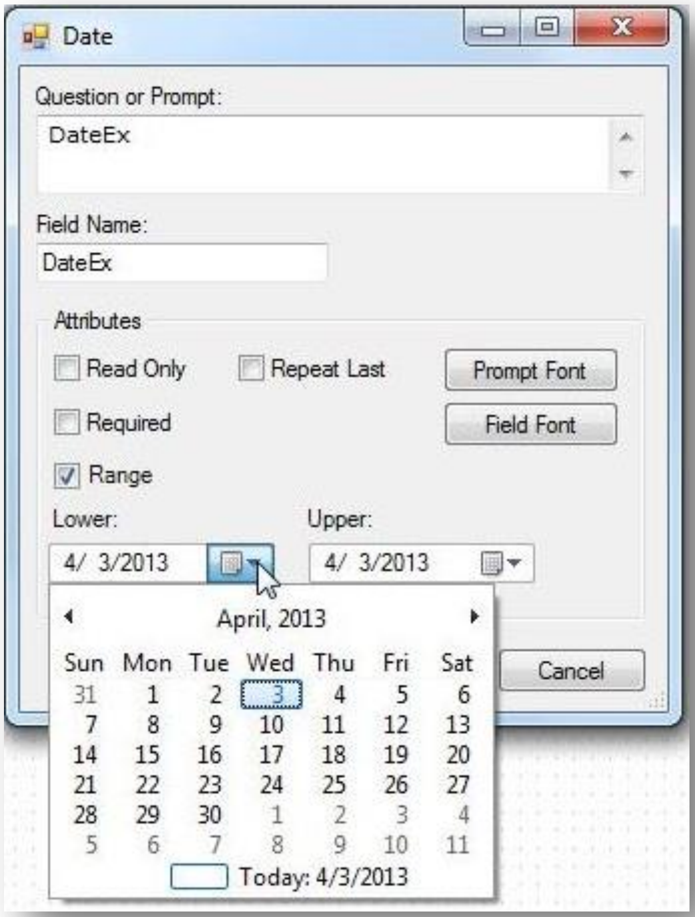


Figure 7.38: Date Field Definition dialog box

5. Click **OK**. The Date field appears on the canvas.

Time

The **Time** field is used to collect time data in the form of hours, minutes, seconds, and AM or PM. The time pattern for data entry is determined by the computer's regional settings where the pattern can be modified. The time pattern cannot be changed within Epi Info™ 7.

To add a Time field:

1. Open the **Time Field Definition** dialog box.
2. Enter the **Question** or **Prompt**. The text entered in this field will display on the canvas and prompt the user to enter a response.

3. Click in the **Field Name** text box or press the tab key. Epi Info automatically suggests a field name based on the Question or Prompt, however, it is very important that field names be short, intuitive, and usable. The field name is used for data validation in Check Code and when doing analyses. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the field name at this time. Field names cannot be changed after data collection starts.

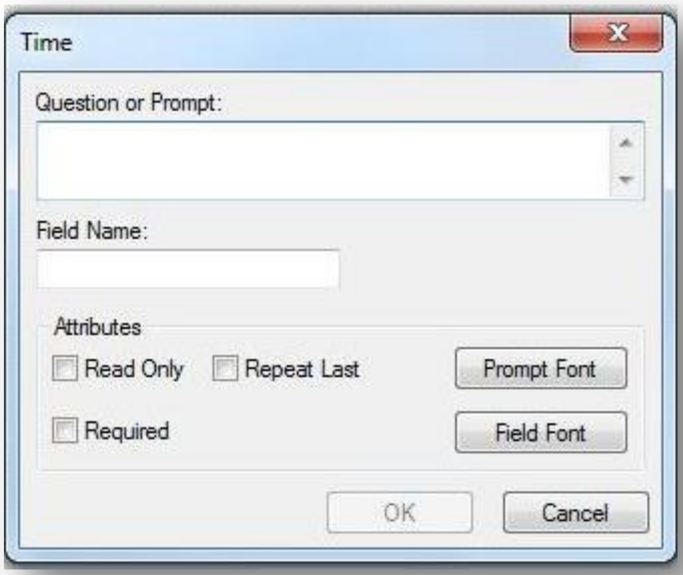
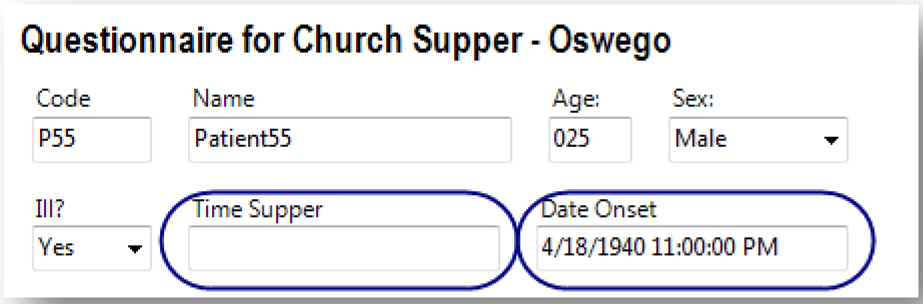


Figure 7.39: Time Field Definition dialog box

4. Click **OK**. The Time field appears on the canvas.

Date-Time

The **Date-Time** field is used to collect data representing the exact date and time of a given moment. It is like a date field combined with a time field. The Date-Time pattern for data entry is determined by the computer's regional settings where the pattern can be changed, if needed. The date-time pattern cannot be changed within Epi Info™ 7. This field type is not supported on Web Survey. The following figure provides an example of how a Date-Time field appears in Enter (circled in blue) based on the Oswego form of the Sample project.



Questionnaire for Church Supper - Oswego

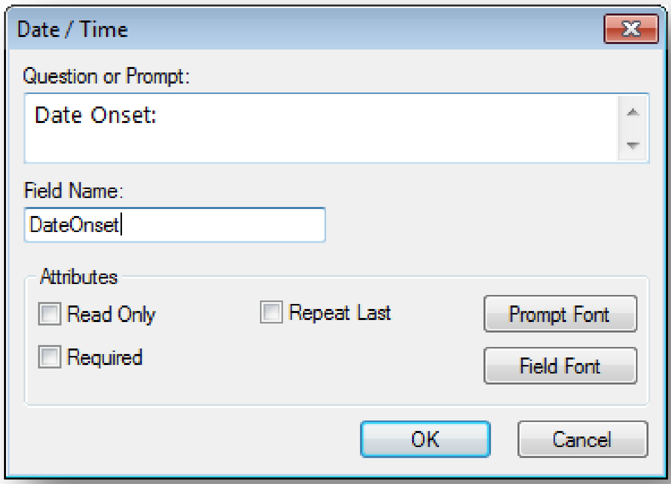
Code: P55
Name: Patient55
Age: 025
Sex: Male
Ill?: Yes
Time Supper:
Date Onset: 4/18/1940 11:00:00 PM

Figure 7.40: Date-Time field

To add a Date-Time field:

1. Open the **Date-Time Field** Definition dialog box.
2. Enter the **Question** or **Prompt**. The text entered in this field will display on the canvas and prompt the user to enter a response.
3. Click in the **Field Name** text box or press the tab key. Epi Info automatically suggests a field name based on the Question or Prompt, however, it is very important that field names be short, intuitive, and usable. The field name is used for data validation in Check Code and when doing analyses. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the field name at this time. Field names cannot be changed after data collection starts.



Date / Time

Question or Prompt: Date Onset:

Field Name: DateOnset

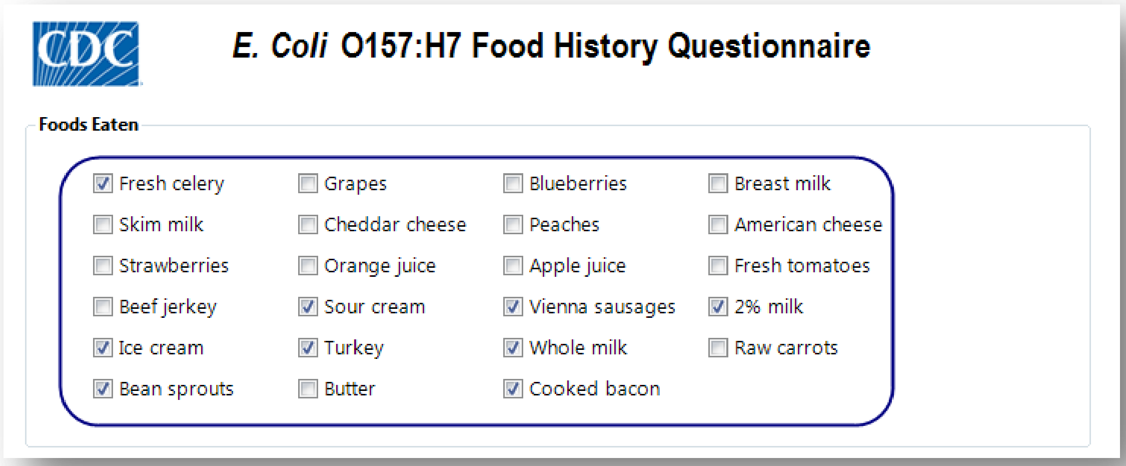
Attributes
- Read Only
- Repeat Last
- Required

Figure 7.41: Date-Time Field Definition dialog box

4. Click **OK**. The Date-Time field appears on the canvas.

Checkbox

The **Checkbox** field allows you to collect data by checking or unchecking a box. Multiple checkboxes can be used to quickly enter responses that are consistent in the study (e.g. symptoms or foods eaten). Checkbox fields collect binary data such as 0 or 1, True or False, Yes or No. The response is stored in the database as a 1 or 0 where 1 equals Yes and 0 equals No. When writing Check Code, use (+) and (–) to indicate yes and no, respectively. Unlike the Yes-No field type, Checkbox fields do not contain missing values, so all checkbox fields are considered to be "No" until they are checked during data entry or assigned a "Yes" value in Check Code. Since checkbox fields can only be either Yes or No, these fields cannot be required fields. The following figure provides an example of how Checkbox fields appear in Enter (circled in blue) based on the FoodHistory form of the EColi project.



E. Coli O157:H7 Food History Questionnaire

Foods Eaten

[✓ Fresh celery]	Grapes	Blueberries	Breast milk
Skim milk	Cheddar cheese	Peaches	American cheese
Strawberries	Orange juice	Apple juice	Fresh tomatoes
Beef jerkey	[✓ Sour cream]	[✓ Vienna sausages]	[✓ 2% milk]
[✓ Ice cream]	[✓ Turkey]	[✓ Whole milk]	Raw carrots
[✓ Bean sprouts]	Butter	[✓ Cooked bacon]	

Figure 7.42: Checkbox field

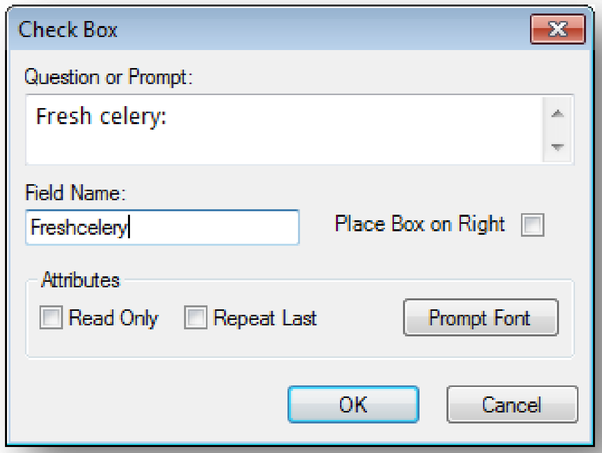
To add a Checkbox field:

1. Open the **Checkbox Field Definition** dialog box.

2. Enter the **Question** or **Prompt**. The text entered in this field will display on the canvas and prompt the user to enter a response.

3. Click in the **Field Name** text box or press the tab key. Epi Info automatically suggests a field name based on the Question or Prompt, however, it is very important that field names be short, intuitive, and usable. The field name is used for data validation in Check Code and when doing analyses. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the field name at this time. Field names cannot be changed after data collection starts.



Question or Prompt: Fresh celery:

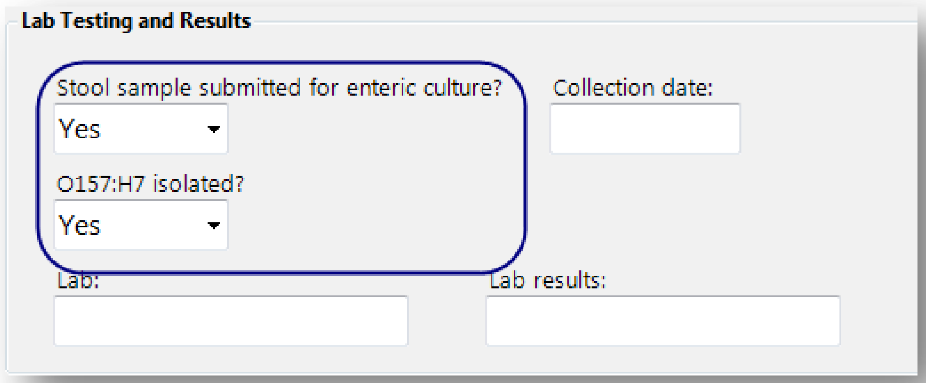
Field Name: Freshcelery

Figure 7.43: Check Box Field Definition dialog box

4. Click **OK**. The Checkbox field appears on the canvas.

Yes-No

The **Yes-No** field is used to collect data with only a Yes or No answer. The field appears as a drop-down list on the canvas. The answer is stored in a database as a 1 or 0 where 1 = Yes and 0 = No. When writing Check Code, use (+) and (–) to indicate yes and no, respectively. Until this field is answered one way or the other, the field's value is considered to be missing. In Check Code, the missing condition is indicated by the (.) operator. The following figure provides an example of how a Yes-No field appears in Enter (circled in blue) based on the FoodHistory form of the EColi project.



Stool sample submitted for enteric culture? Yes

O157:H7 isolated? Yes

Figure 7.44: Yes-No field

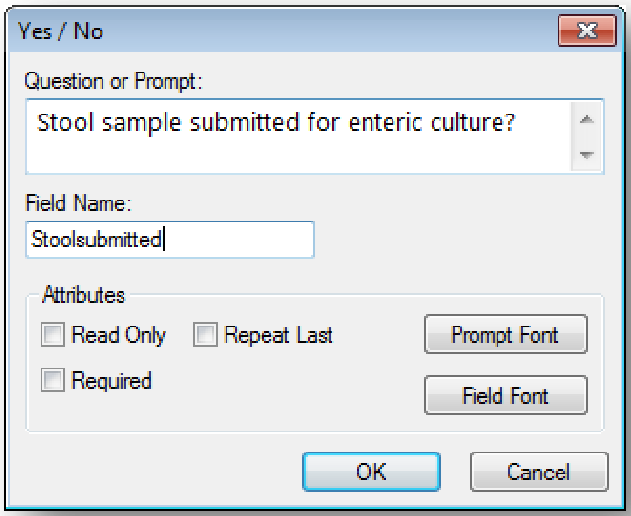
In the Enter module and in the output of analyses, the values for Yes-No fields can be displayed in a variety of ways, or customized for a specific purpose such as in translation to another language. Regardless of how the data are displayed, the values in the database and in exported output are still only 1 or 0.

Yes-No fields can display their data as:		
Yes	No	Missing
True	False	Unknown
(+)	(-)	(.)
<custom setting>	<custom setting>	<custom setting>

To add a Yes-No field:

1. Open the **Yes-No Field Definition** dialog box.
2. Enter the **Question** or **Prompt**. The text entered in this field will display on the canvas and prompt the user to enter a response.
3. Click in the **Field Name** text box or press the tab key. Epi Info automatically suggests a field name based on the Question or Prompt, however, it is very important that field names be short, intuitive, and usable. The field name is used for data validation in Check Code and when doing analyses. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the field name at this time. Field names cannot be changed after data collection starts.



Question or Prompt: Stool sample submitted for enteric culture?

Field Name: Stoolsubmitted

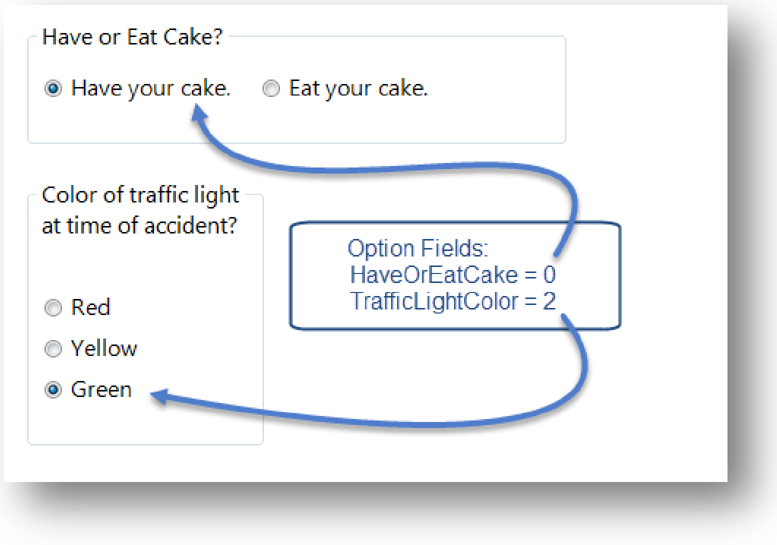
Read Only ☐ Repeat Last ☐ Required ☐

Figure 7.45: Yes-No Field Definition dialog box

4. Click **OK**. The Yes-No field appears on the canvas.

Option

The **Option** field creates a group box containing mutually exclusive option buttons, also known as radio buttons. These are used for fields requiring one and only one answer. When a choice is made, any other selection within the group is cleared. In the database, the values are stored as a zero-based array. The first option is 0, the second is 1, the third is 2, and so on. The image below shows an example of two option fields. Since the options are mutually exclusive, in the first example you can't select both options "Have your cake" and "Eat your cake". You may select only one or the other. The field name for this example option field is "HaveOrEatCake". Since the first option is selected, the field "HaveOrEatCake" has the value of zero (0). In the next example, the field name is "TrafficLightColor". Since the selected color is the third option, the value of "TrafficLightColor" is two (2).



Option Fields:
HaveOrEatCake = 0
TrafficLightColor = 2

Figure 7.46: Option Field

To add an Option field:

1. Open the **Option Field Definition** dialog box.

2. Enter the **Question** or **Prompt**. The text entered in this field will display on the canvas and prompt the user to enter a response.

3. Click in the **Field Name** text box or press the tab key. Epi Info automatically suggests a field name based on the Question or Prompt, however, it is very important that field names be short, intuitive, and usable. The field name is used for data validation in Check Code and when doing analyses. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the field name at this time. Field names cannot be changed after data collection starts.

4. Select **Vertical** or **Horizontal** from the **List options vertically or horizontally** field to choose how the options are displayed in the Option field.

5. In the **Start listing options on the left or right** field, select Left or Right to choose where the options are listed.

6. In the **Attributes** field, select **Right** or **Left** to determine how the radio buttons are aligned with the option text.

7. Select the **Number of Choices** in the **Attributes** field. You can have a maximum of 16 options. The number of options selected corresponds to the number of rows listed under **Option Definition**.

8. Enter the option name in each row of the **Option Definition** section.

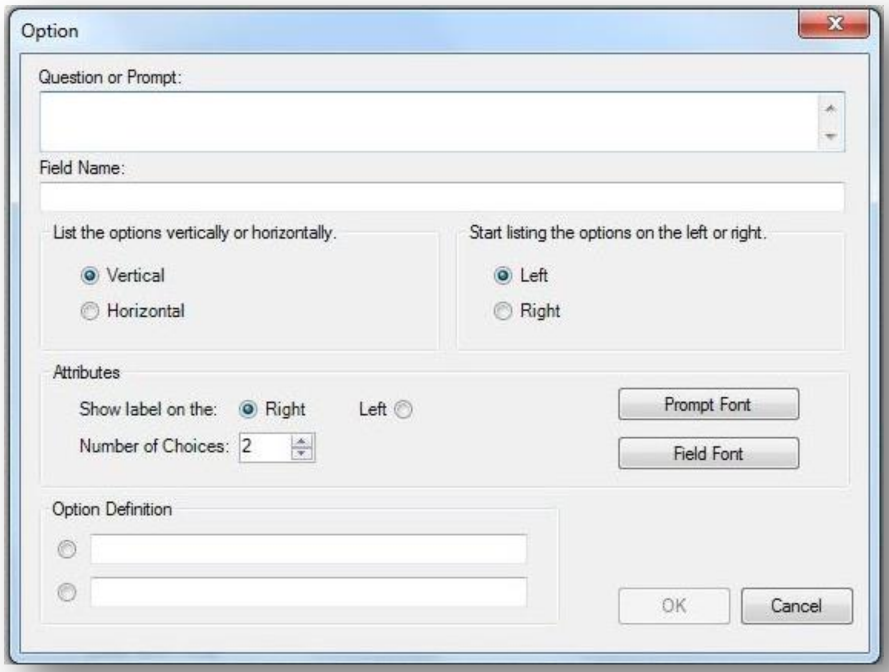


Figure 7.47: Option Field Definition dialog box

9. Click **OK**. The Option field appears on the canvas.

Command Button

The **Command Button** creates a clickable button on the form and is used to run a specific block of Check Code. Below are a few examples of the possible applications of a Command Button and the Check Code behind the button:

- Compare the values of fields and perform automatic calculations
- Raise a message for information or to alert the data entry person to invalid data
- Capture longitude and latitude coordinates from an address field
- Run a prewritten program in Classic Analysis

- Open the Visual Dashboard with a saved canvas
- Open other programs outside of Epi Info™ 7 such as Microsoft Excel or Outlook.

The Command Button field is not supported on Web Survey. The following figure provides an example of how a Command Button appears in Enter (circled in blue) based on the FoodHistory form of the EColi project.

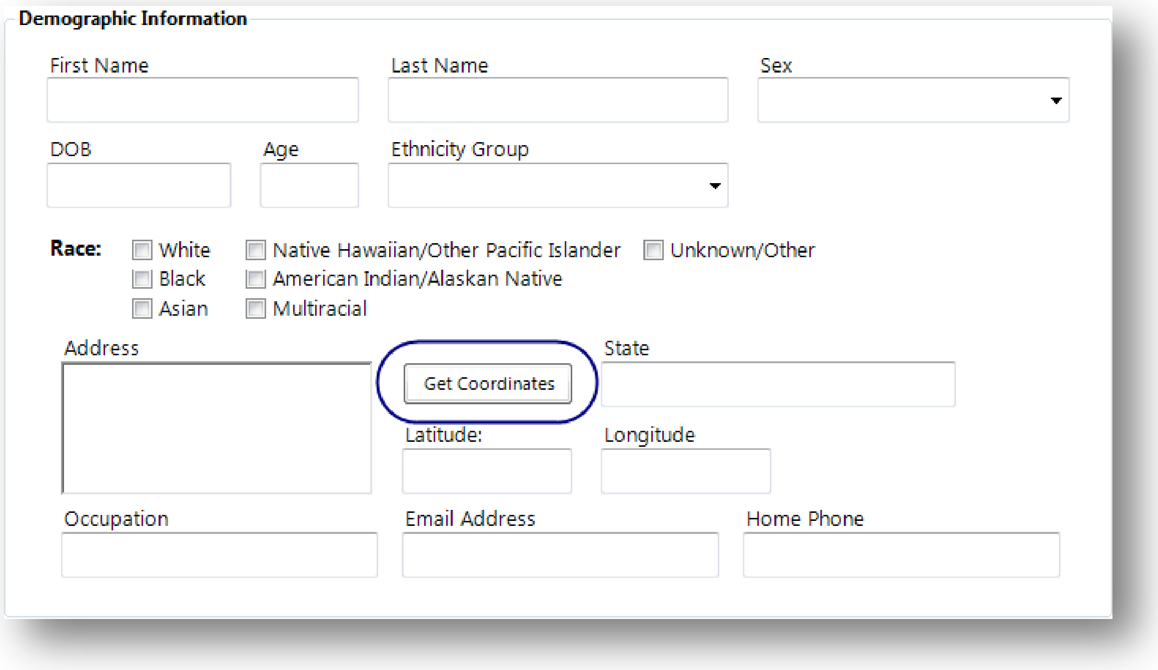
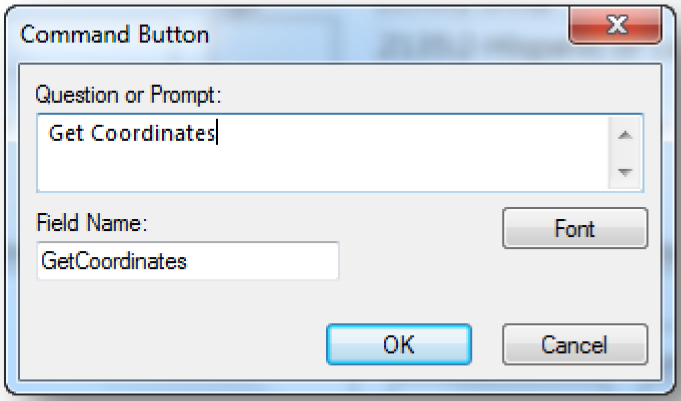


Figure 7.48: Command Button

To add a Command Button field:

1. Open the **Command Button Field Definition** dialog box.
2. Enter the **Question** or **Prompt**. The text entered in this field will display on the canvas and prompt the user to enter a response.
3. Click in the **Field Name** text box or press the tab key. Epi Info automatically suggests a field name based on the Question or Prompt, however, it is very important that field names be short, intuitive, and usable. The field name is used for data validation in Check Code and when doing analyses. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the field name at this time. Field names cannot be changed after data collection starts.



Get Coordinates

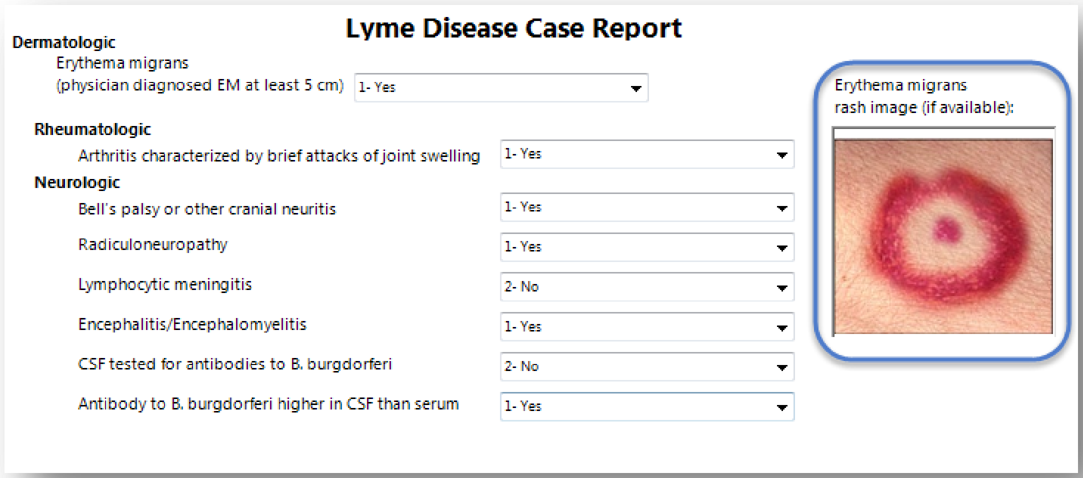
GetCoordinates

Figure 7.49: Command Button Field Definition dialog box

4. Click **OK**. The Command Button field appears on the canvas.

Image

The **Image** field allows an image to be inserted in the record. Examples of items that could be included as an image might be the patient's photo, an image of a wound, rash or insect bite, a bacteria culture dish, or the barcode of a sample vial. This is especially useful when the data collection device has a built in camera such as a tablet computer and smart phone. If the image is not collected from a built in camera, clicking in this field will display the Open dialog to browse for an image file. The accepted image file types are: Portable Network Graphics (.PNG), Graphics Interchange Format (.GIF), Joint Photographic Expert Group (.JPG or .JPEG), and Windows Bitmap Format (.BMP). This field type is not supported on Web Survey.



Lyme Disease Case Report

Dermatologic

Erythema migrans
(physician diagnosed EM at least 5 cm): 1- Yes

Rheumatologic

Arthritis characterized by brief attacks of joint swelling	1- Yes

Neurologic

Bell's palsy or other cranial neuritis	1- Yes
Radiculoneuropathy	1- Yes
Lymphocytic meningitis	2- No
Encephalitis/Encephalomyelitis	1- Yes
CSF tested for antibodies to B. burgdorferi	2- No
Antibody to B. burgdorferi higher in CSF than serum	1- Yes

Erythema migrans
rash image (if available):

Figure 7.50: Image field

To add an Image field:

1. Open the **Image Field Definition** dialog box.
2. Enter the **Question** or **Prompt**. The text entered in this field will display on the canvas and prompt the user to enter a response.
3. Click in the **Field Name** text box or press the tab key. Epi Info automatically suggests a field name based on the Question or Prompt, however, it is very important that field names be short, intuitive, and usable. The field name is used for data validation in Check Code and when doing analyses. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the field name at this time. Field names cannot be changed after data collection starts.

4. By default, Epi Info™ expands/contracts the image to fit in the size of the Image field. Check the **Retain Image Size** checkbox to disable this function and have the image inserted in the field without modification.

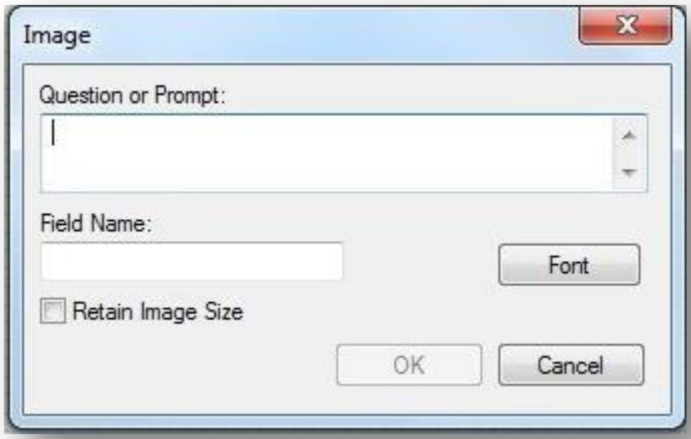


Figure 7.51: Image Field Definition dialog box

5. Click **OK**. The Image field appears on the canvas.

Mirror

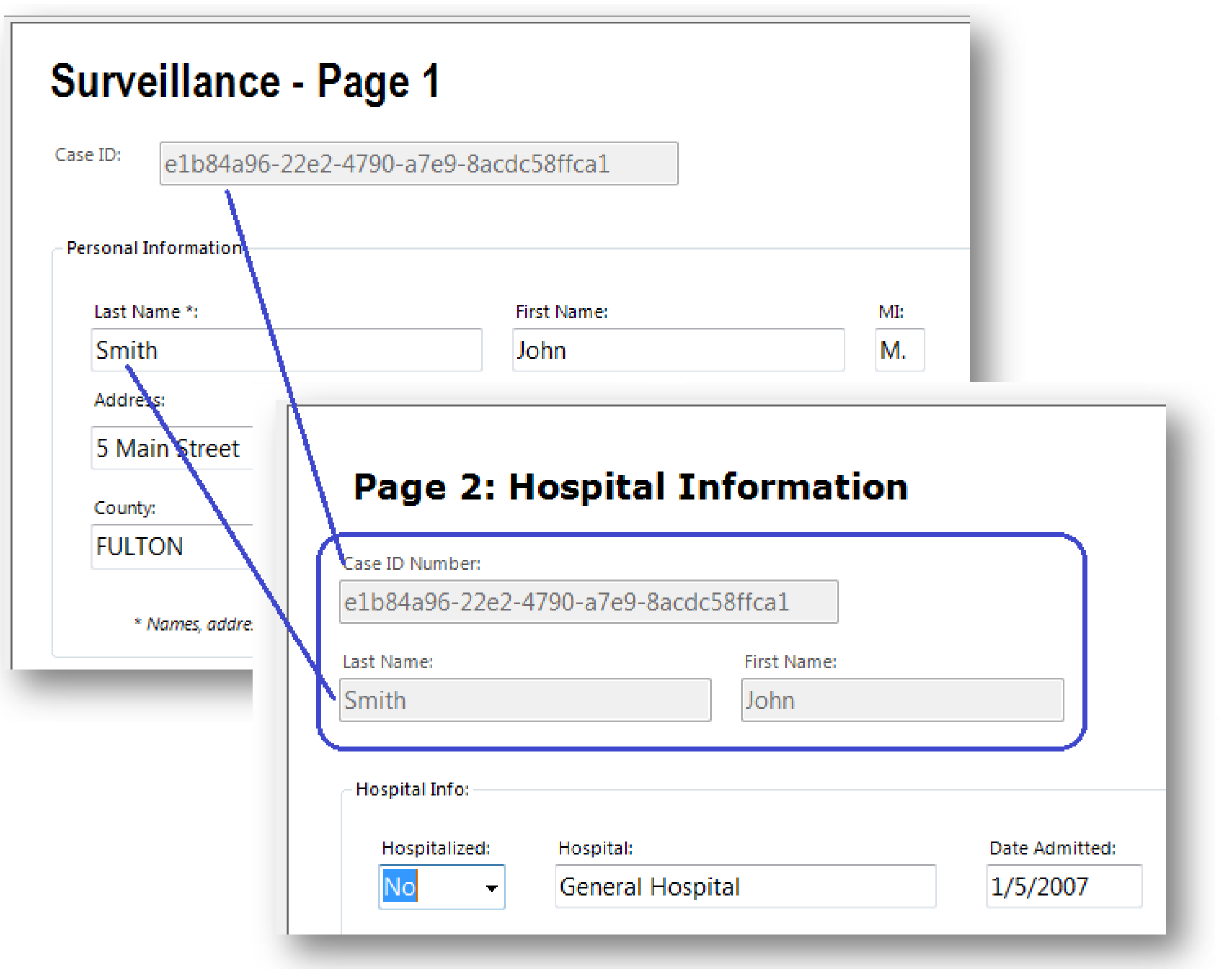
The **Mirror** field shows data in a field on another page. It does not actually store data of its own. It only shows data from another field. This is useful when the same information needs to be displayed on multiple pages in a form. Since this field type does not allow any data entry, it does not have a column in the database, Check Code cannot be run on this field, and Mirror fields are not searchable. This field type is not supported on Web Survey.

Mirror fields also have the following attributes:

- Mirror fields are Read Only.

- When the source field receives data through data entry or when it is assigned a value in Check Code, the value of that field will be reflected in the Mirror field.
- Mirror fields can be copied and pasted to subsequent pages while in the Form Designer. However, since they are Read Only, their contents cannot be copied when running the form in the Enter tool.
- Command buttons cannot be selected as source fields to mirror.

The following figure is an example showing three mirror fields on Page 2 of a form that reflect the data entered into Page 1 of the form. The mirror fields on Page 2 are circled in blue.


Surveillance - Page 1

Case ID: e1b84a96-22e2-4790-a7e9-8acdc58ffca1

Personal Information

Last Name *: Smith

First Name: John

MI: M.

Address: 5 Main Street

County: FULTON

* Names, addre

Page 2: Hospital Information

Case ID Number: e1b84a96-22e2-4790-a7e9-8acdc58ffca1

Last Name: Smith

First Name: John

Hospital Info:

Hospitalized: No

Hospital: General Hospital

Date Admitted: 1/5/2007

Figure 7.52: Mirror fields reflecting data entered on Page 1

To add a Mirror field:

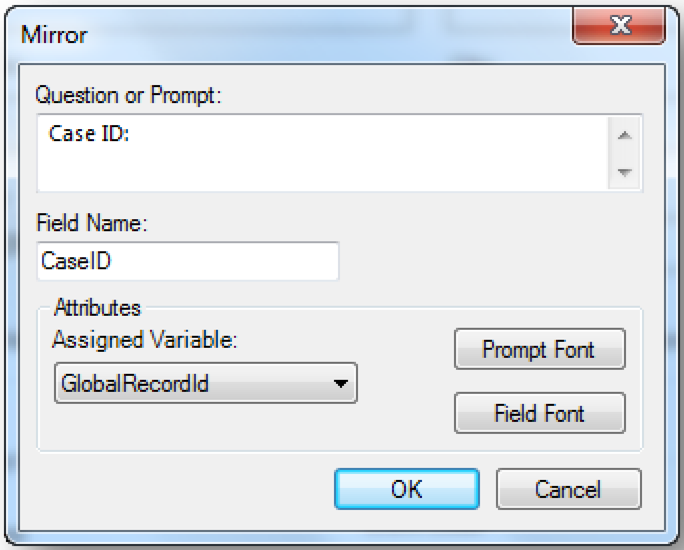
1. Open the **Mirror Field Definition** dialog box.
2. Enter the **Question** or **Prompt**. The text entered in this field will display on the canvas and prompt the user to enter a response.

3. Click in the **Field Name** text box or press the tab key. Epi Info automatically suggests a field name based on the Question or Prompt, however, it is very important that field names be short, intuitive, and usable. The field name is used for hiding or unhiding mirror fields in Check Code and other commands. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the field name at this time. Field names cannot be changed after data collection starts.

4. Click the **Assigned Variable** drop-down list in the Attributes Group to show a list of variables that can be mirrored.

5. Select the **variable** to be mirrored.



Question or Prompt: Case ID:

Field Name: CaseID

Attributes

Assigned Variable: GlobalRecordId

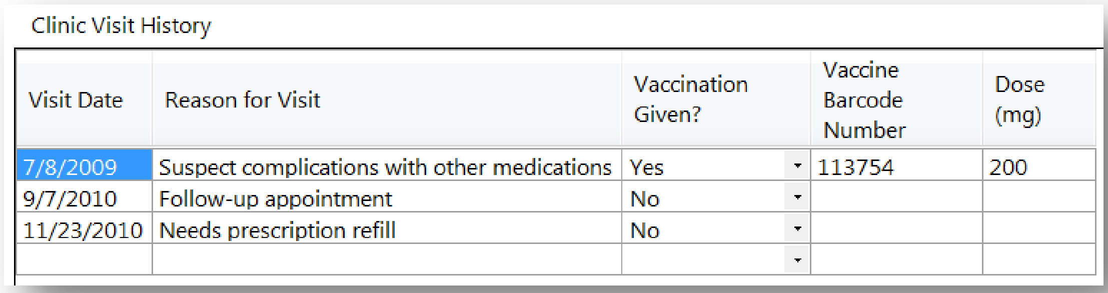
Figure 7.53: Mirror Field Definition dialog box

6. Click **OK**. The Mirror field appears on the canvas.

Grid

The Grid field allows users to enter data in the form of a spreadsheet on the page. This allows for simple and effective data entry for multiple data points.

When you create a Grid field, you specify each column for the grid, the type of data for the column, its size and properties.


Clinic Visit History

Visit Date	Reason for Visit	Vaccination Given?	Vaccine Barcode Number	Dose (mg)
7/8/2009	Suspect complications with other medications	Yes	113754	200
9/7/2010	Follow-up appointment	No		
11/23/2010	Needs prescription refill	No		

Figure 7.54a: Grid Field

Before deciding to use a Grid field on your form, please also know the following limitations with Grid fields:

1. Not all field types are supported as grid columns. Text, Number, Date, Time, Date-Time, Phone Number, Checkbox, Yes-No, Legal Value, and Comment Legal field types are supported.

2. Grid fields do not support Check Code. This means that the many things that Check Code can offer, such as automatic calculations, skip patterns, enabling/disabling of fields, or other data entry validation, are not available to Grid fields.

3. Grid column tab order cannot be changed. Pressing the tab key moves the cursor to the next field from left to right. Pressing the Enter key moves the cursor within the same column, but down to the next row.

4. Grid fields are not supported on Web Survey.

To add a Grid field:

1. Open the **Grids Field Definition** dialog box.

2. Enter the **Question** or **Prompt**. The text entered in this field will display on the canvas and prompt the user to enter a response.

3. Click in the **Field Name** text box or press the tab key. Epi Info automatically suggests a field name based on the Question or Prompt, however, it is very important that field names be short, intuitive, and usable. The field name is used for data validation in Check Code and when doing analyses. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the field name at this time. Field names cannot be changed after data collection starts.

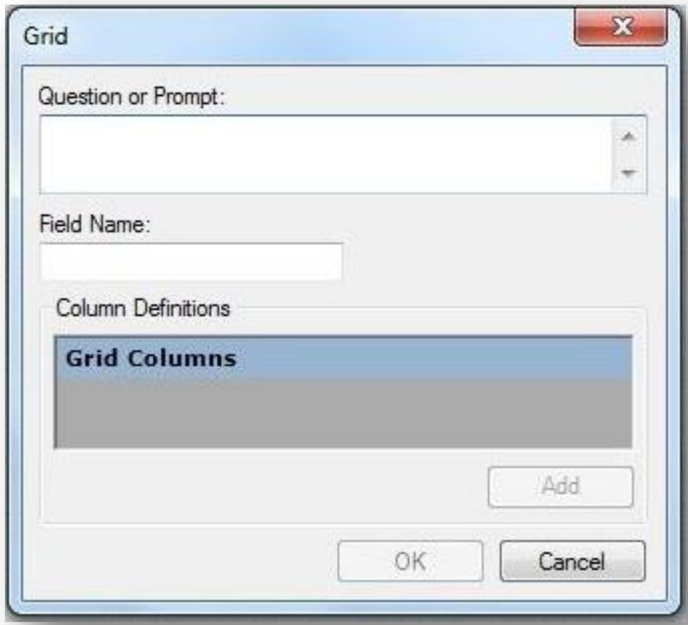


Figure 7.55: Grids Field Definition dialog box

After setting the Question or Prompt for the Grid field and setting it's Field Name, the Add button is enabled.

4. Click the **Add** button.

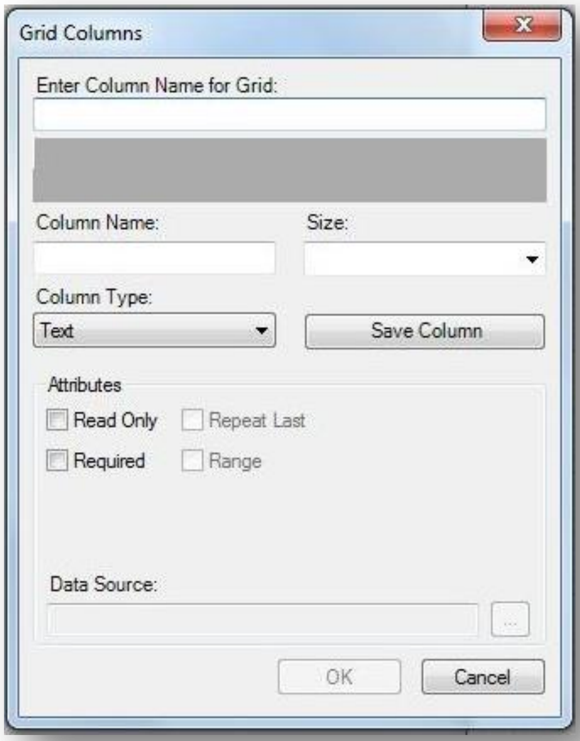


Figure 7.56: Grid Columns

5. Type the column heading into the **Enter Column Name for Grid** textbox. This text will display as the column heading on the form.

6. Press **Tab** or click in the **Column Name** textbox. Epi Info automatically suggests a column name based on the Enter Column name for Grid field, however, it is very important that column names be short, intuitive, and usable. The column name cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the column name at this time. Column names cannot be changed after data collection starts.

7. Select the column size from the **Size** drop-down list. This determines the amount of characters that can be placed in that column's cells. This setting only applies to Text type columns.

8. Select the column format from the **Column Type** drop-down list. Available formats are:

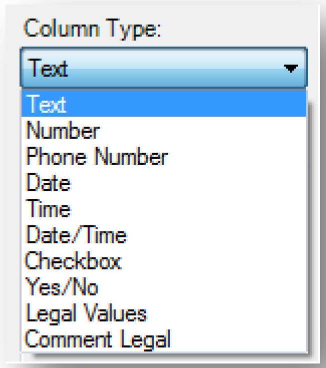


Figure 7.57: Grid Column Types

9. After selecting a Legal Values or Comment Legal column type, click the browse button next to the Data Source field and complete the steps outlined in the Legal Values and Comment Legal field sections below.
10. Click **Save Column** to add the column to the grid.
11. Repeat steps 5-10 until all desired columns have been added.
12. Click **OK**. The Grid field appears on the canvas.

Legal Values

A Legal Values field creates a drop-down list of choices on the field. The responses listed in the drop-down are the only choices available to the respondent and cannot be altered. The following figure provides an example of how Legal Value fields appear in Enter (circled in blue) based on the Surveillance form of the Sample project.

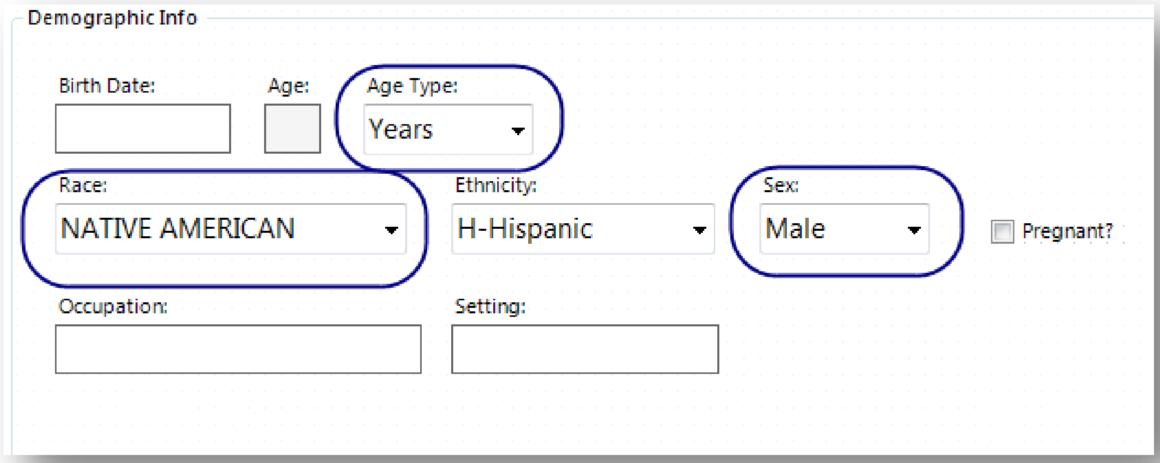


Figure 7.58: Legal Values field

To add a Legal Values field:

1. Open the **Legal Values Field Definition** dialog box.
2. Enter the **Question** or **Prompt**. The text entered in this field will display on the canvas and prompt the user to enter a response.
3. Click in the **Field Name** text box or press the tab key. Epi Info automatically suggests a field name based on the Question or Prompt, however, it is very important that field names be short, intuitive, and usable. The field name is used for data validation in Check Code and when doing analyses. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the field name at this time. Field names cannot be changed after data collection starts.

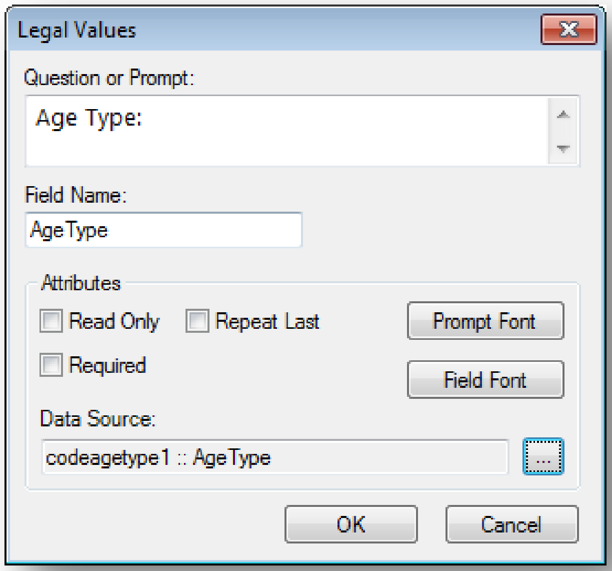
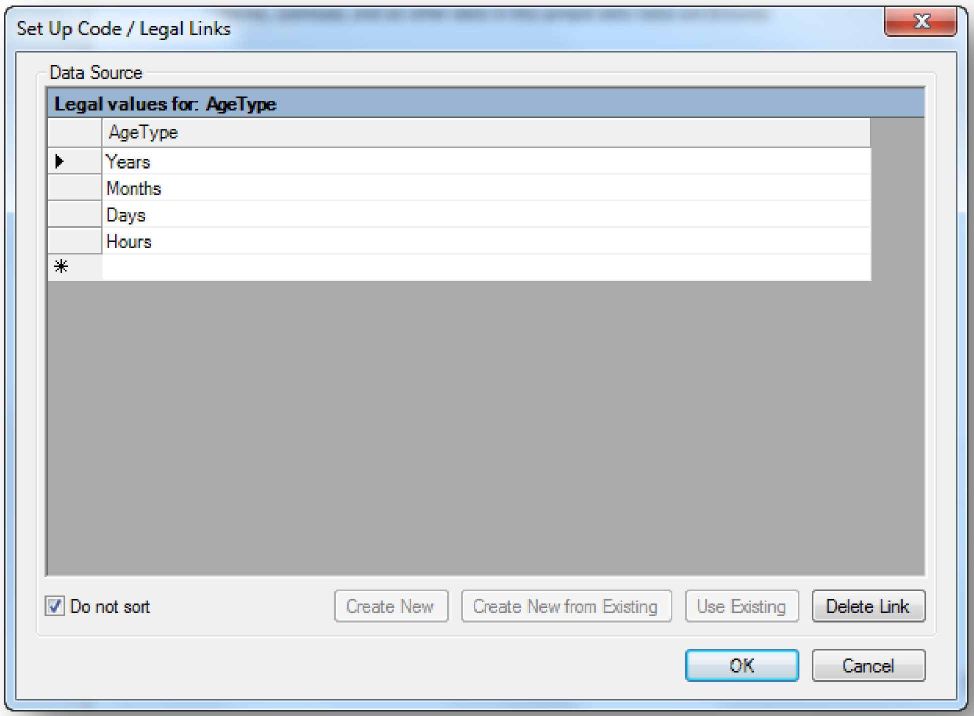


Figure 7.59: Legal Values Field Definition dialog box

4. Click the **Browse** button to the right of the Data Source textbox.
5. Click **Create New** to enter the available answers for the question.



Legal values for: AgeType

AgeType
Years
Months
Days
Hours

Figure 7.60: Set Up Code / Legal Links

6. Enter the first value (e.g. Years). Press **Enter** or **Tab** to advance to the next value.
7. Enter additional values until all necessary items are added to the drop-down list.
8. Values will appear in alphabetical order unless you select **Do Not Sort.** This feature displays values in the order in which they were entered.
9. Click **OK**.
10. Existing tables can also be used to create legal values.
    1. Click **Create New from existing** (takes a copy of selected table, changes to the original table will not affect the new table) or **Use Existing** (establishes a connection between the selected table and the new table, any changes made to the original table will modify the new table)
    2. Select a table from the drop-down list.
    3. Click **OK**.
11. From the Field Definition box, click **OK**. The new field appears in the form as a drop-down list of values.

Comment Legal

Comment Legal fields are similar to Legal Values. They are text fields with character(s) typed in front of the text (with a hyphen). During data entry, the character and the text (e.g., M-Male) are displayed. However, only the character value is stored in the table (e.g., M). In the Classic Analysis and Visual Dashboard tools, statistics can be calculated if all values are numeric. The following figure provides an example of how a Comment Legal field appears in Enter (circled in blue) based on the CaseReport form of the Lyme project.

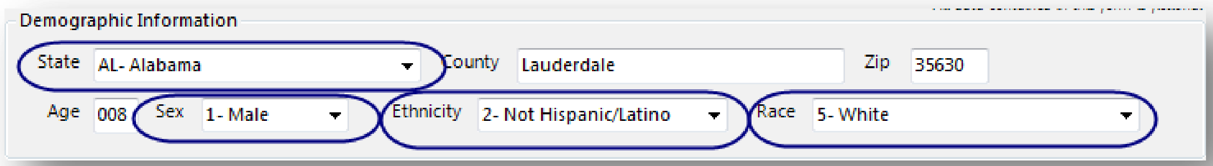

Demographic Information
State: AL- Alabama
County: Lauderdale
Zip: 35630
Age: 008
Sex: 1- Male
Ethnicity: 2- Not Hispanic/Latino
Race: 5- White

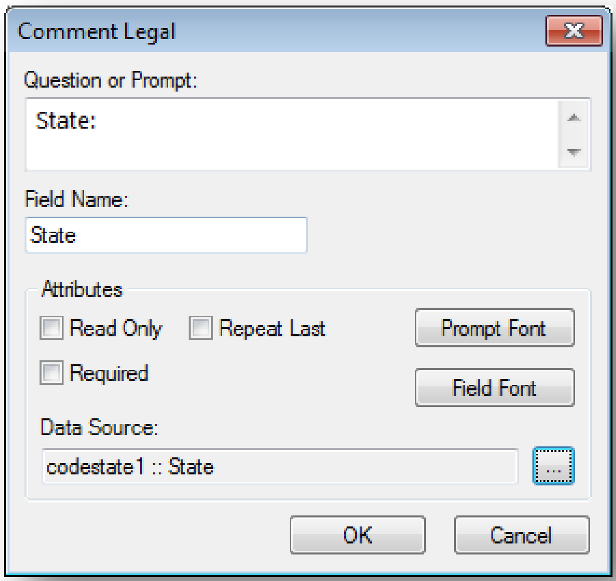
Figure 7.61: Comment Legal field

To add a Comment Legal field:

1. Open the **Comment Legal Values Field Definition** dialog box.
2. Enter the **Question** or **Prompt**. The text entered in this field will display on the canvas and prompt the user to enter a response.

3. Click in the **Field Name** text box or press the tab key. Epi Info automatically suggests a field name based on the Question or Prompt, however, it is very important that field names be short, intuitive, and usable. The field name is used for data validation in Check Code and when doing analyses. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the field name at this time. Field names cannot be changed after data collection starts.



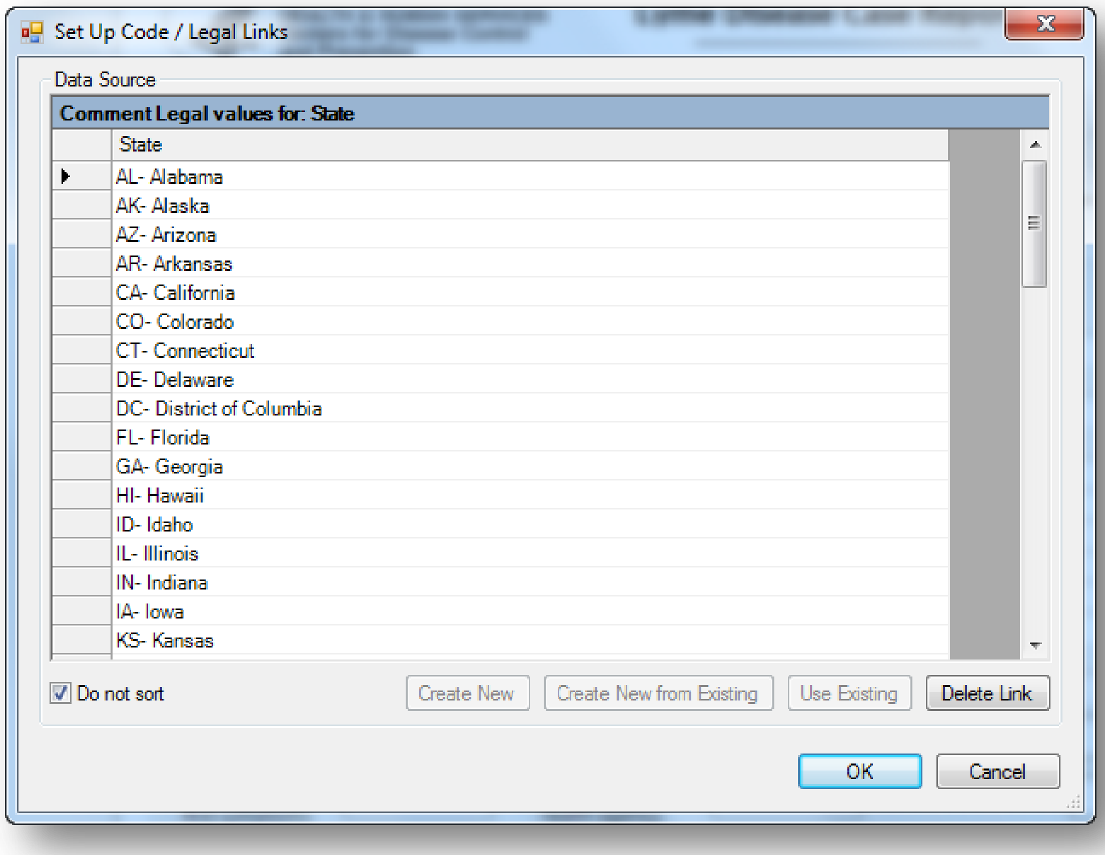
Question or Prompt: State:

Field Name: State

Data Source: codestate1 :: State

Figure 7.62: Comment Legal Field Definition dialog box

4. Click the **browse** button to the right of the Data Source textbox.
5. Click **Create New** to enter the available answers for the question.


Set Up Code / Legal Links

Data Source

Comment Legal values for: State

	State
▸	AL- Alabama
	AK- Alaska
	AZ- Arizona
	AR- Arkansas
	CA- California
	CO- Colorado
	CT- Connecticut
	DE- Delaware
	DC- District of Columbia
	FL- Florida
	GA- Georgia
	HI- Hawaii
	ID- Idaho
	IL- Illinois
	IN- Indiana
	IA- Iowa
	KS- Kansas

☑ Do not sort

Figure 7.63: Set Up Code / Legal Links

6. Enter the first value using the form code-hyphen-comment as in "M-Male" where "M" is the code and "Male" is the comment. Press **Enter** or **Tab** to advance to the next value.

7. Enter additional values using the same code-hyphen-comment format until all necessary items are added to the drop-down list.

8. Values will appear in alphabetical order unless you select **Do Not Sort** which will display values in the order in which they are entered.

9. Click **OK**.

10. Existing tables can also be used to create comment legal fields.

    1. Click **Create New from existing** (takes a copy of selected table, changes to the original table will not affect the new table) or **Use Existing** (establishes a connection between the field and the existing table. Any changes made to the original table will be seen in all fields that use this table as the source.

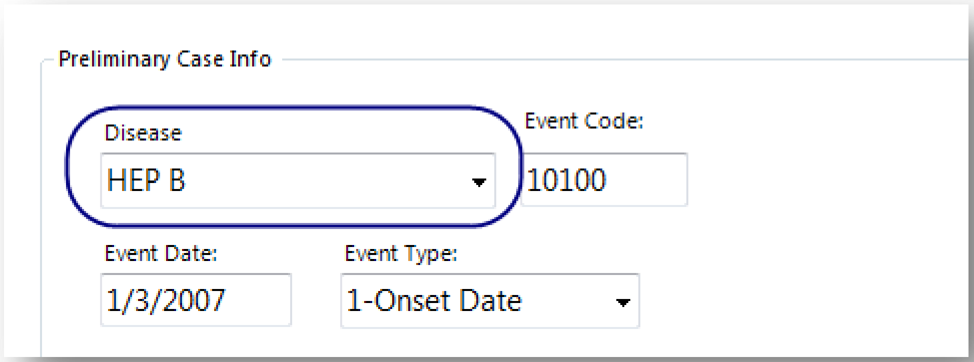
    2. Select a table from the drop-down list.

    3. Click **OK**.

11. From the Field Definition box, click **OK**. The new field appears in the form as a drop-down list of values.

Codes

A Codes field designates the available options on a form based on the user's response to a question. Based on the value selected from a drop-down list, other field(s) are populated with predetermined values. At least two fields must exist; one which holds the selection code, and another to receive the value of the code. The first field holds the selection code in a drop-down list while subsequent fields are Read Only and are populated based on the values in the code table. The Codes field allows for more efficient data entry by the user. This field type is not supported on Web Survey. The following figure provides an example of how a Codes field appears in Enter (circled in blue) based on the Surveillance form of the Sample project.



Disease: HEP B

Event Code: 10100

Event Date: 1/3/2007

Event Type: 1-Onset Date

Figure 7.64: Codes field

Create a New Code Table

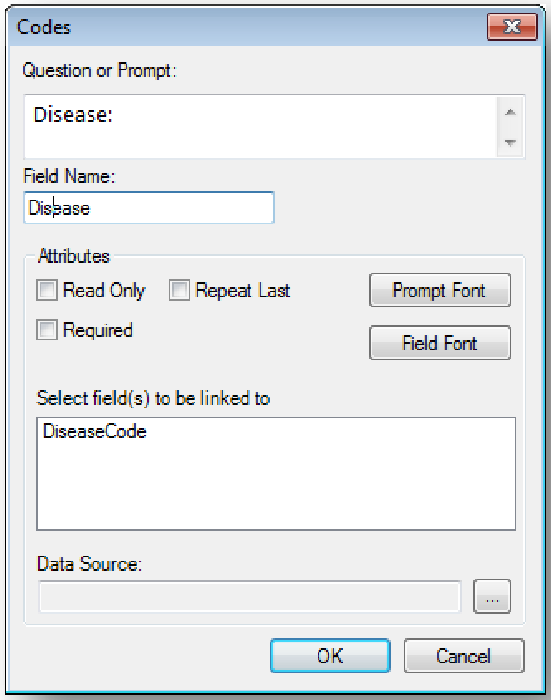
1. Open the **Codes Field Definition** dialog box.

2. Enter the **Question** or **Prompt**. The text entered in this field will display on the canvas and prompt the user to enter a response.

3. Click in the **Field Name** text box or press the tab key. Epi Info automatically suggests a field name based on the Question or Prompt, however, it is very important that field names be short, intuitive, and usable. The field name is used for data validation in Check Code and when doing analyses. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the field name at this time. Field names cannot be changed after data collection starts.

4. Select the **field(s)** to be linked from the **Select field(s) to be linked** section. To select multiple fields, hold down the **CTRL** key and click each **field**.



Question or Prompt: Disease:

Field Name: Disease

Attributes

[ ] Read Only [ ] Repeat Last

[ ] Required

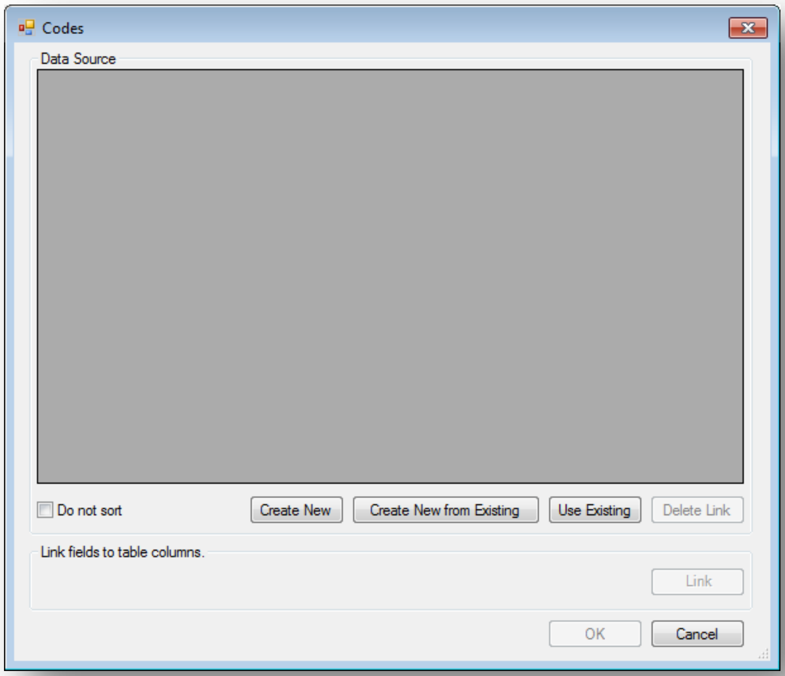
Select field(s) to be linked to

DiseaseCode

Data Source:

Figure 7.65: Codes Field Definition dialog box

5. Click on the **Data Source Browse** button.



[ ] Do not sort

Link fields to table columns.

Figure 7.64 b: Create New Code table

6. Click **Create New**. A spreadsheet opens for you to enter the values for the Codes field and Linked fields.

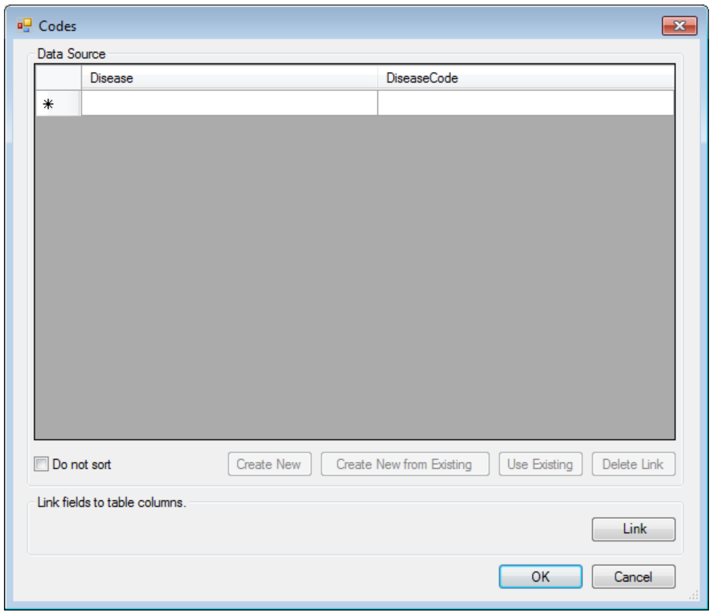
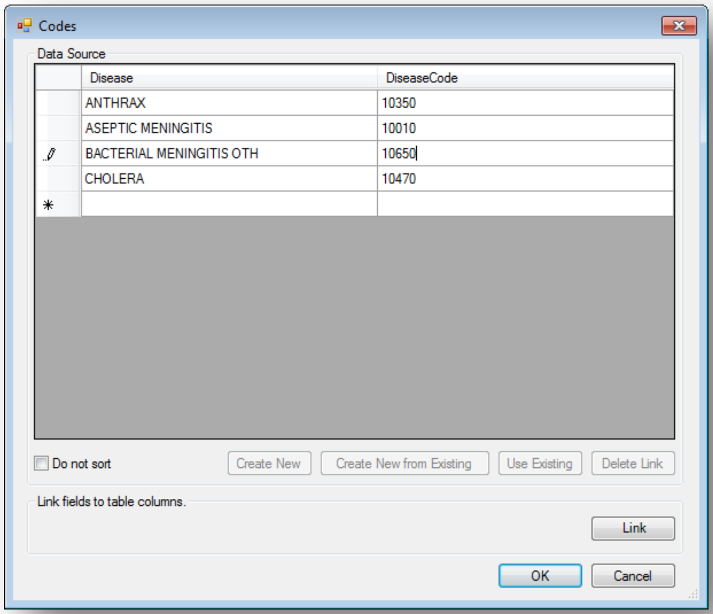


Figure 7.66: Codes Data Source Blank

7. The left-most column displays the selection field(s) chosen in the Fields Definition dialog box.



Disease	DiseaseCode
ANTHRAX	10350
ASEPTIC MENINGITIS	10010
BACTERIAL MENINGITIS OTH	10650
CHOLERA	10470

Figure 7.67: Codes Data Source Complete

8. Each column to the right lists the field(s) to receive the codes based on the value of the selection field.

9. Enter the **codes** for each field.

10. Press the **Tab** key to move to the next field, or to the next row if at the end of a row.

11. Click **OK** to accept the codes for each field.

12. Existing tables can also be used to create code tables.

    1. Click **Create New from Existing** (takes a copy of the selected table, changes to the original table will not affect the new table) or **Use Existing** (establishes a connection between the selected table and the new table, any changes made to the original table will modify the new table)

    2. Select a table from the drop-down list.

    3. Click **OK**.

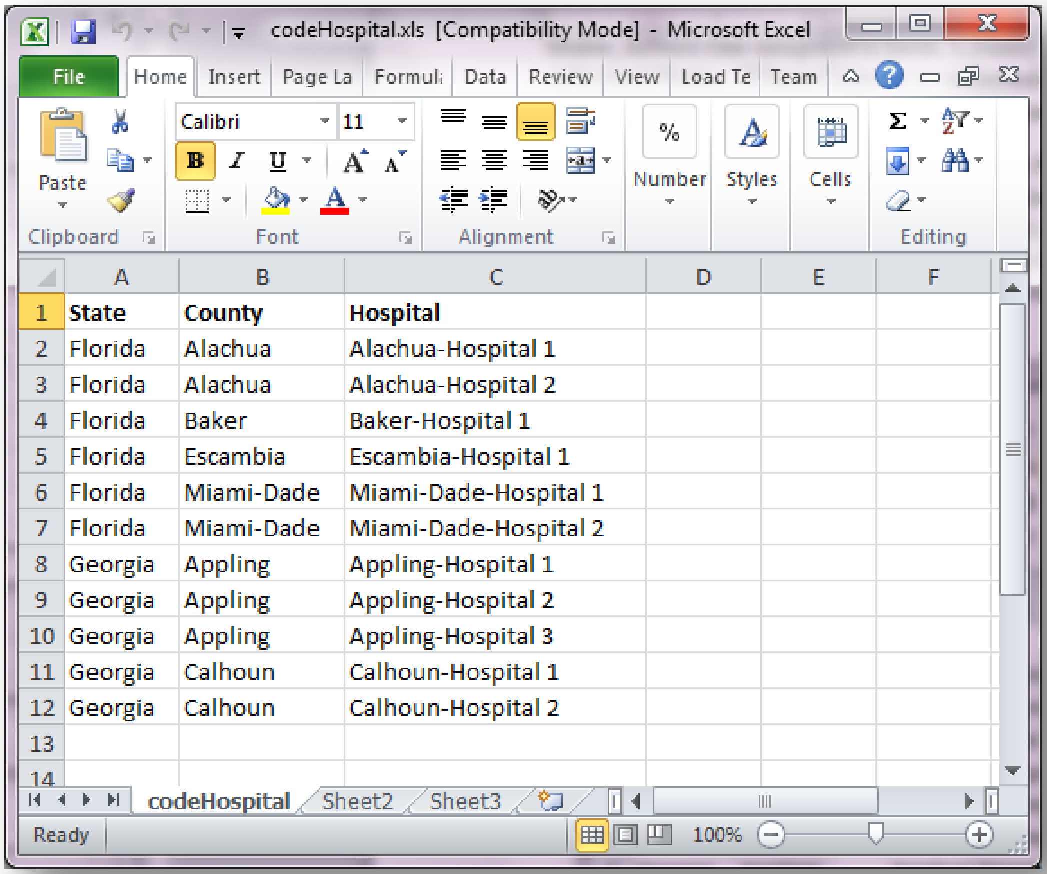
13. Click **OK** to close the Field Definition dialog box and place the fields in the form.

14. To test the code table, open the **Enter** tool and verify that both fields populate based on the drop-down list selection.

Create Cascading Drop-Down Fields

Cascading Drop-Down fields are essentially two or more Code, Legal Value, or Comment Legal fields linked together in such a way as to filter each subsequent field based on the value selected in prior fields. In the example described below, the drop-down field for State, filters the drop-down field, County, to show only those counties that exist in the selected State. When County is selected, the drop-down field for Hospital is shown with only the hospitals located in the selected county.

In order to create Cascading Drop-Down fields, you need to have a table completed with all values for the linked fields that can be imported into Epi Info™ 7. The Excel spreadsheet shown below, named ‘codeHospital’, is an example of such a table. When importing this table into your Epi Info™ 7 project, the table name must contain the prefix “code”, as in this example, “**code**Hospital”.



	A	B	C
1	**State**	**County**	**Hospital**
2	Florida	Alachua	Alachua-Hospital 1
3	Florida	Alachua	Alachua-Hospital 2
4	Florida	Baker	Baker-Hospital 1
5	Florida	Escambia	Escambia-Hospital 1
6	Florida	Miami-Dade	Miami-Dade-Hospital 1
7	Florida	Miami-Dade	Miami-Dade-Hospital 2
8	Georgia	Appling	Appling-Hospital 1
9	Georgia	Appling	Appling-Hospital 2
10	Georgia	Appling	Appling-Hospital 3
11	Georgia	Calhoun	Calhoun-Hospital 1
12	Georgia	Calhoun	Calhoun-Hospital 2

Figure 7.68: Cascading Drop-down Fields

To add Cascading Drop-down fields, complete the following steps:

1. Add the first **Codes** field to the page, but don't link the field to other fields, yet. You will do this in later steps. For this example, we named the field StateName.

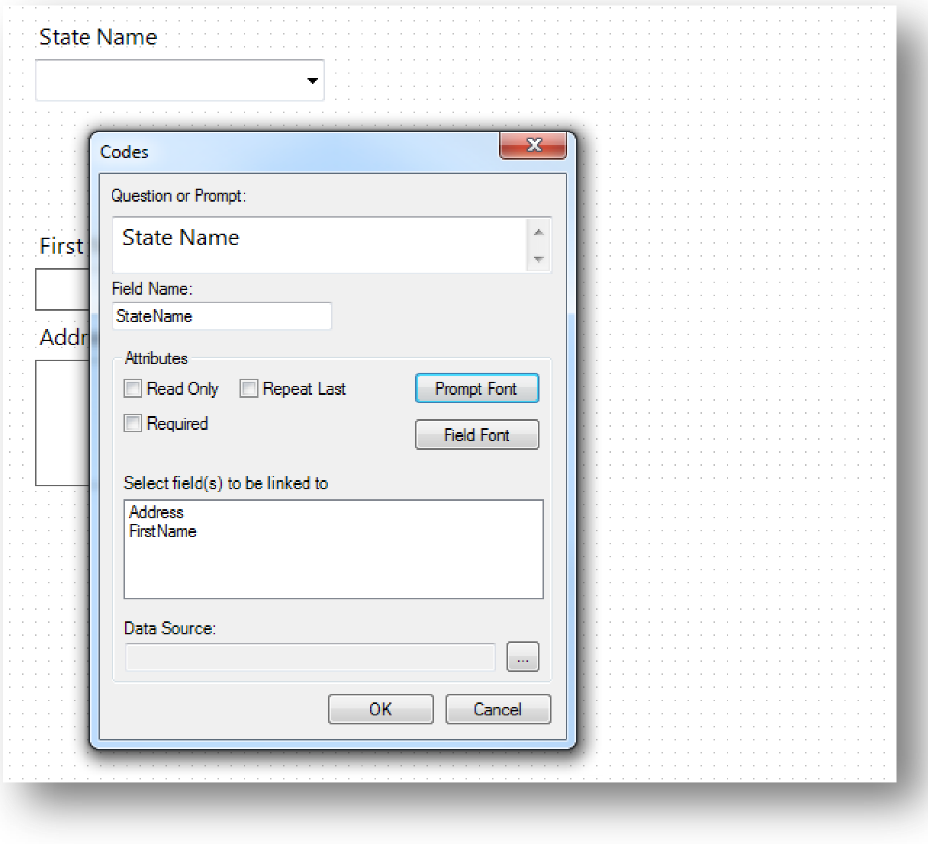


Figure 7.69: Adding Codes to Cascading Drop-down Fields

2. Add another **Codes** field to the page. Again, don't link the field to other fields, yet. In this example, we named the second field County. Repeat this step for all but the last field in your cascading sequence. In this example, we have only three fields to link, so the next field will be our last.

3. Add the last **field** in our cascading sequence. The last field should be a type other than a Codes field, such as Legal Value or Comment Legal. Since no other fields will be linked to this last field, you don't want to use a Codes field. In our example, we have two Codes fields named 'StateName' and 'County', and one Legal Value field named 'Hospital'.

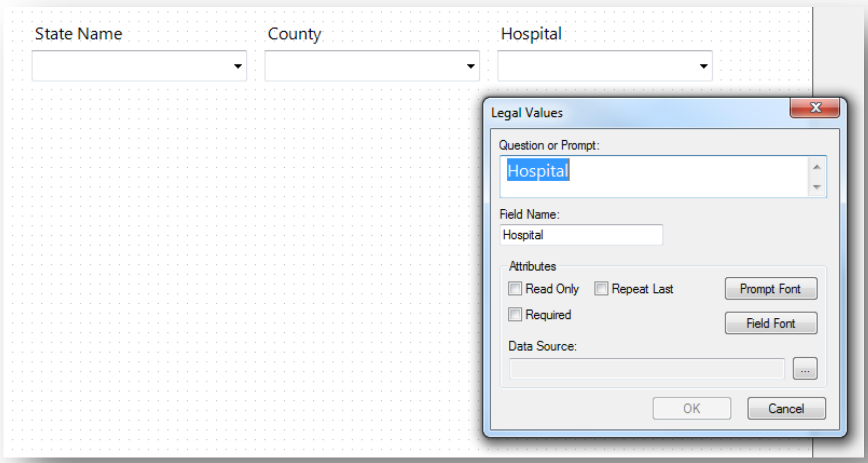


Figure 7.70: Adding Legal Values to Cascading Drop-down Fields

4. If using a Legal Value or Comment Legal field for the last field, click the browse button to the right of Data Source. The Set Up Code / Legal Links dialog opens.

5. Click **Use Existing**. The Open Form – Select a Table dialog box opens.

6. Select the code table that was imported and click **OK**. In this example, we selected the codeHospital table.

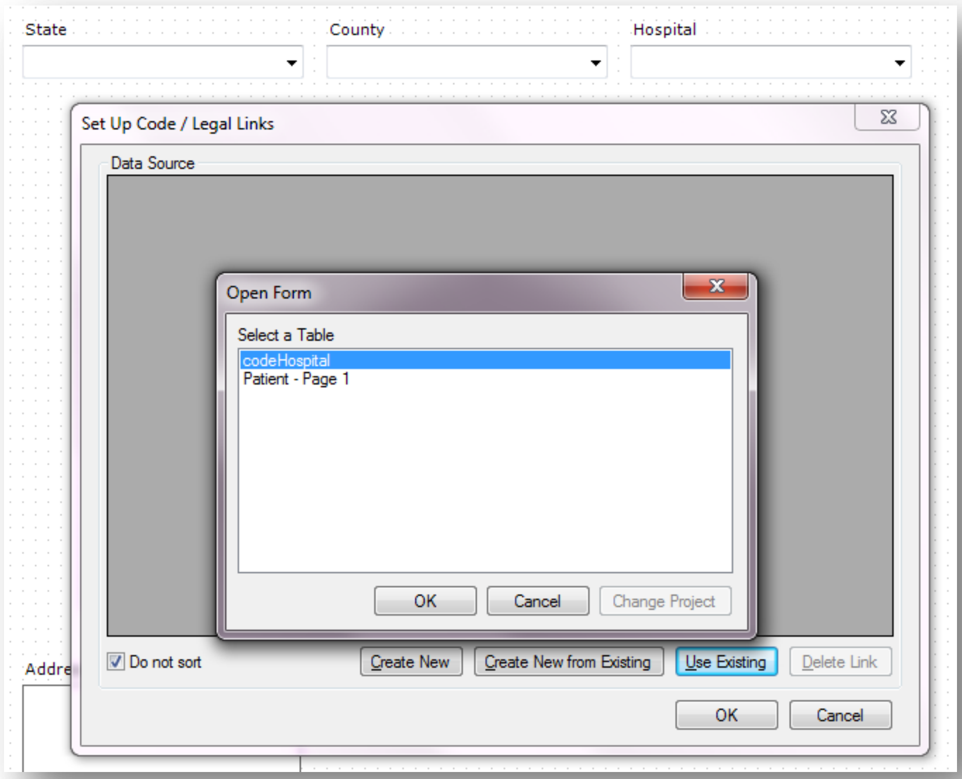


Figure 7.71: Open Form Dialog Box

7. Click **OK**. The Select a field dialog opens showing the columns in the codeHospital table that can be linked.

8. Select the appropriate **field**. Because we are linking the Hospital field, we selected Hospital in this example.

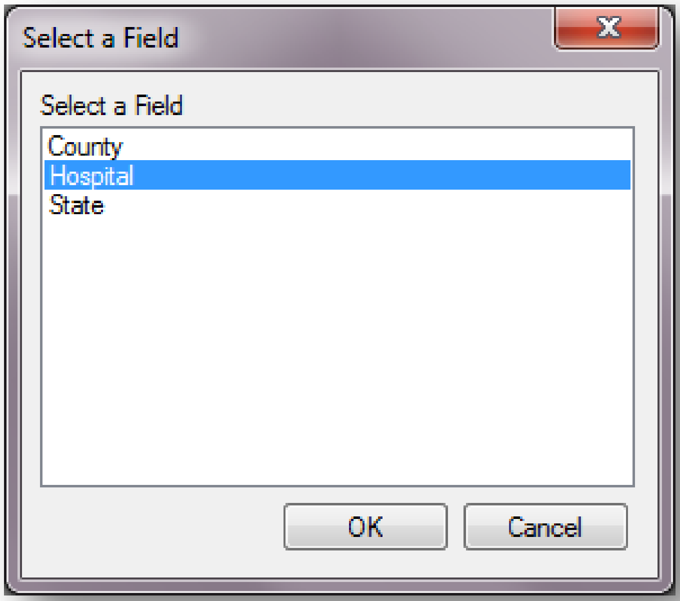
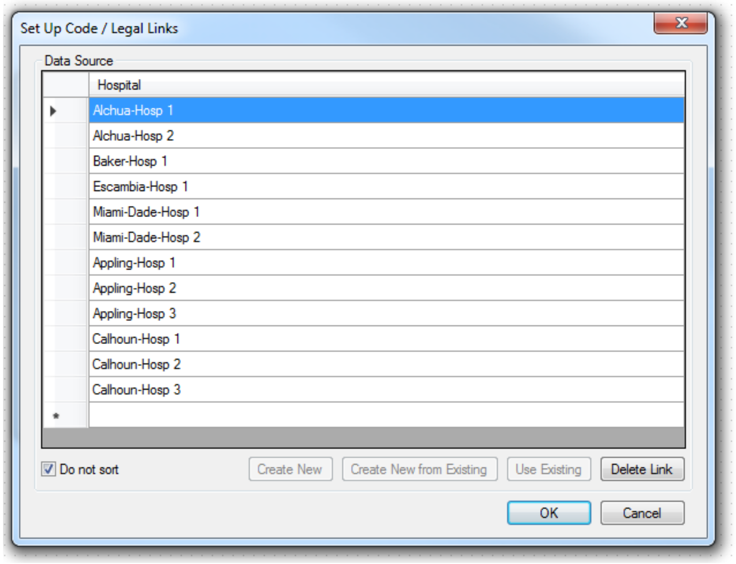


Figure 7.72: Selecting a Field

9. Click **OK** on the **Select a Field** dialog. The Set Up Code / Legal Links data source table shows the values from the Hospital column in table.

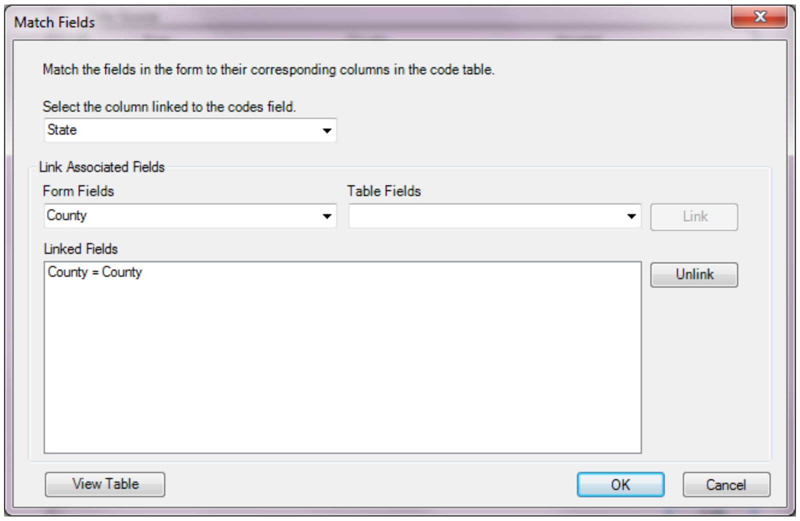


Hospital
Alchua-Hosp 1
Alchua-Hosp 2
Baker-Hosp 1
Escambia-Hosp 1
Miami-Dade-Hosp 1
Miami-Dade-Hosp 2
Appling-Hosp 1
Appling-Hosp 2
Appling-Hosp 3
Calhoun-Hosp 1
Calhoun-Hosp 2
Calhoun-Hosp 3

Figure 7.73a: Data Source shows values for Hospital

10. Click **OK** on the **Set Up Code / Legal Links** dialog box.

11. Click **OK** on the field definition dialog. The next steps are to link the fields to the remaining Codes fields.
12. **Open** the properties for the first **Codes** field added above. (Right click the field and select **Properties**.) Our first field was 'StateName' and we want to link this field to the County field so when the State Name is selected, only the corresponding counties are shown in the County field.
13. In the **Select Fields to be Linked** box, select the field you want to be filtered. Since, in our example, we want the County field to be filtered based on the selection made for State, we should select **County** here.
14. Click the **Browse** button to the right of **Data Source**. The Codes dialog opens.
15. Click **Use Existing**. The **Select a Table** dialog box opens.
16. Select the same code table used to link the last field. In our example, **codeHospital** is the code table we are using.
17. Click **Link**. The **Match Fields** dialog box opens. This is where we match the fields in our cascading sequence to the columns in the code table.
18. Select the **column to be linked** to the first **Codes** field. Since our first field is StateName, we selected the column from our imported table named '**State**'.
19. In the **Link Associated Fields** group, select the **Form Field to be linked**. For our example we want County to be linked to State, so we selected **County**.
20. In the **Table Fields** drop-down, select the corresponding column in the code table.
21. Click **Link**. The association to the linked field appears in the Linked Fields box.


Match Fields

Match the fields in the form to their corresponding columns in the code table.

Select the column linked to the codes field.
State

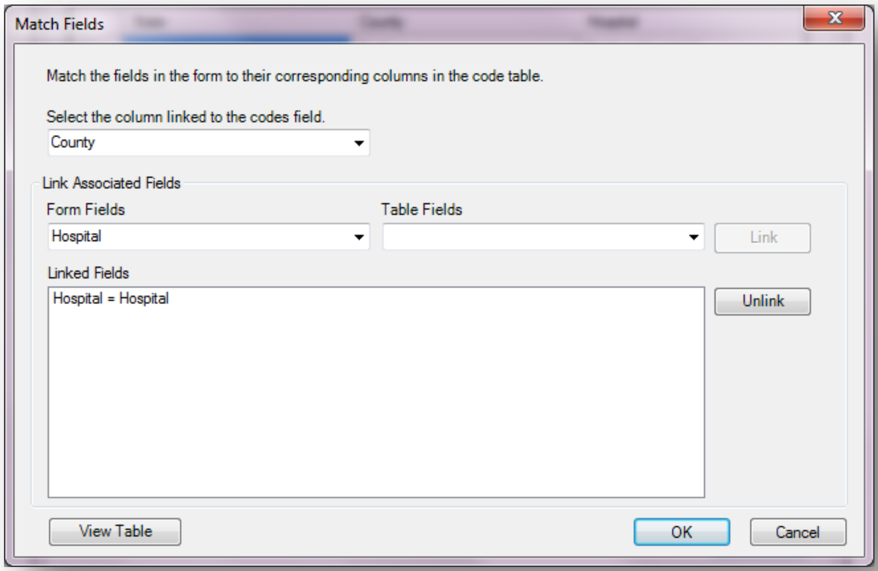
Link Associated Fields
Form Fields: County
Table Fields:

Linked Fields
County = County

Figure 7.74: Match Fields Dialog Box

22. Click **OK** to close the Match Fields dialog.

23. Repeat the steps 12 through 21 for each remaining Codes field in the cascading sequence. For each, specify the next field in the cascading sequence and link it to the corresponding field in the code table. In our example, County is the only remaining Codes field to be linked and this field will be linked to Hospital.



Match the fields in the form to their corresponding columns in the code table.

Select the column linked to the codes field.
County

Form Fields: Hospital

Linked Fields:
Hospital = Hospital

Figure 7.75: Linked Fields

This completes the process to create a series of cascading drop-down fields. When the form is opened in Enter, the selection made in the first field in the cascading sequence will filter the values shown in the next field in the sequence. In our example, when Florida is selected for State Name, the list shown for County only includes the values Alchua, Baker, Escambia, and Miami-Dade.

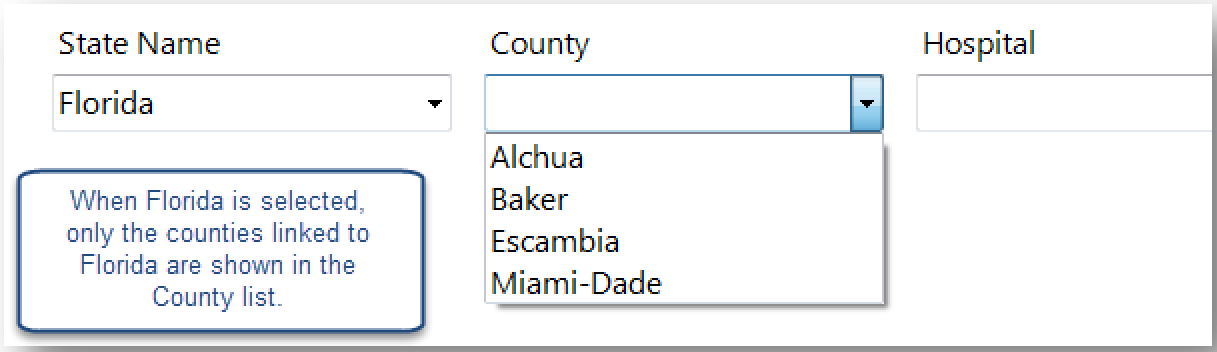
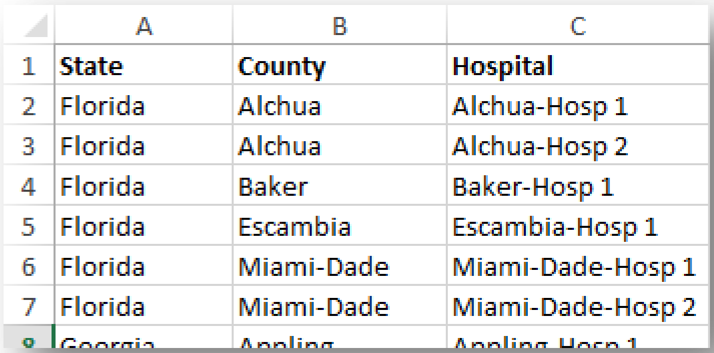

State Name
Florida
County
Alchua
Baker
Escambia
Miami-Dade
Hospital
When Florida is selected, only the counties linked to Florida are shown in the County list.

Figure 7.74: State Name field now filters the values shown for the County field

The StateName codes field filters the County codes field because of they way they are linked and because Alchua, Baker, Escambia, and Miami-Dade are the only counties on the same rows as Florida in the codeHospital table.



	A	B	C
1	**State**	**County**	**Hospital**
2	Florida	Alchua	Alchua-Hosp 1
3	Florida	Alchua	Alchua-Hosp 2
4	Florida	Baker	Baker-Hosp 1
5	Florida	Escambia	Escambia-Hosp 1
6	Florida	Miami-Dade	Miami-Dade-Hosp 1
7	Florida	Miami-Dade	Miami-Dade-Hosp 2
8	Georgia	Appling	Appling-Hosp 1

Figure 7.765: The codeHosptal table imported from Excel.

Likewise, when the county Alchua is selected, only the values Alchua-Hosp1 and Alchua-Hosp2 are shown in the Hospital legal values field.

Relate

Related forms are relationships between the main or parent form with sub or child forms. They are used to establish a one-to-one or one-to-many relationship between parent form records and child form records. For example, a survey may have a form devoted to patient information. This may be related to a form to record the many hospital visits that each patient may have. Related forms are linked to a parent form automatically by unique keys generated by Epi Info™ 7. In Enter, the related form is opened by clicking a Related Form Button. When a Related Form Button is selected, it will open the first page of the related form. Using Check Code, the Relate Button can be made conditionally accessible (e.g., to show a special form for a particular disease). This field type is not supported on Web Survey or on the Companion for Android application. The following figure is an example of how a Relate field appears in Enter (circled in blue) based on the Surveillance form of the Sample project.

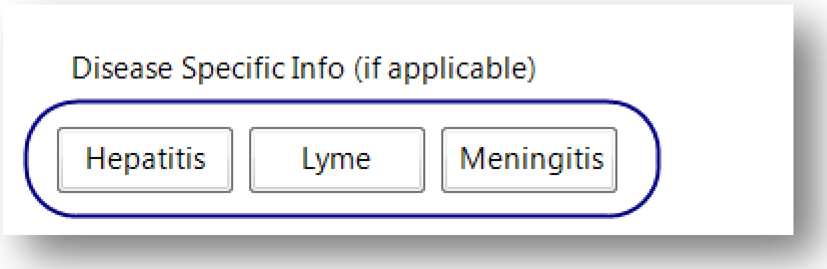


Figure 7.77: Relate field

To add a Relate field:

1. **Open** the **Relate Field Definition** dialog box.
2. Enter the **Question** or Prompt. The text entered in this field will display on the canvas and prompt the user to enter a response.
3. Click in the **Field Name** text box or press the **tab key**. Epi Info automatically suggests a field name based on the Question or Prompt, however, it is very important that field names be short, intuitive, and usable. The field name is used for data validation in Check Code and when doing analyses. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the field name at this time. Field names cannot be changed after data collection starts.

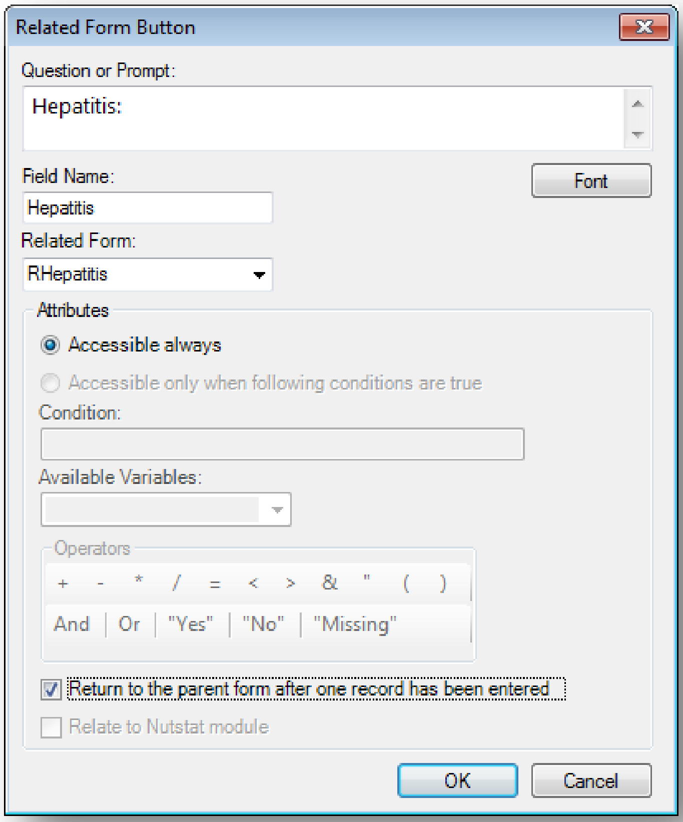


Figure 7.78: Relate Form Button Field Definition dialog box

4. From the Related Form drop-down list, select **Create new form** to create a new form or select an existing form.

    - The dialog box will show the **Accessible always** button selected. This option cannot be modified. The **Accessible only when following conditions are true** selection is not available in this version of Epi Info™ 7.

    - **Accessible always** will create a related button in the form that is active at all times during data entry.

    - **Accessible only when following conditions are true** will be available in a future Epi Info™ release. However, a conditional statement can be created using Check Code (see Check Code). Check Code can be associated with the button to create a condition statement.

5. Select **'Return to the parent form after one record has been entered'** to allow only one record to be entered in the related form for each record in the parent form. In Enter, after a record is saved in the related form, you are returned to the parent form. If multiple entries are desired, then leave this option unselected.

6. Click **OK**. The Related Form button appears on the canvas.

Group

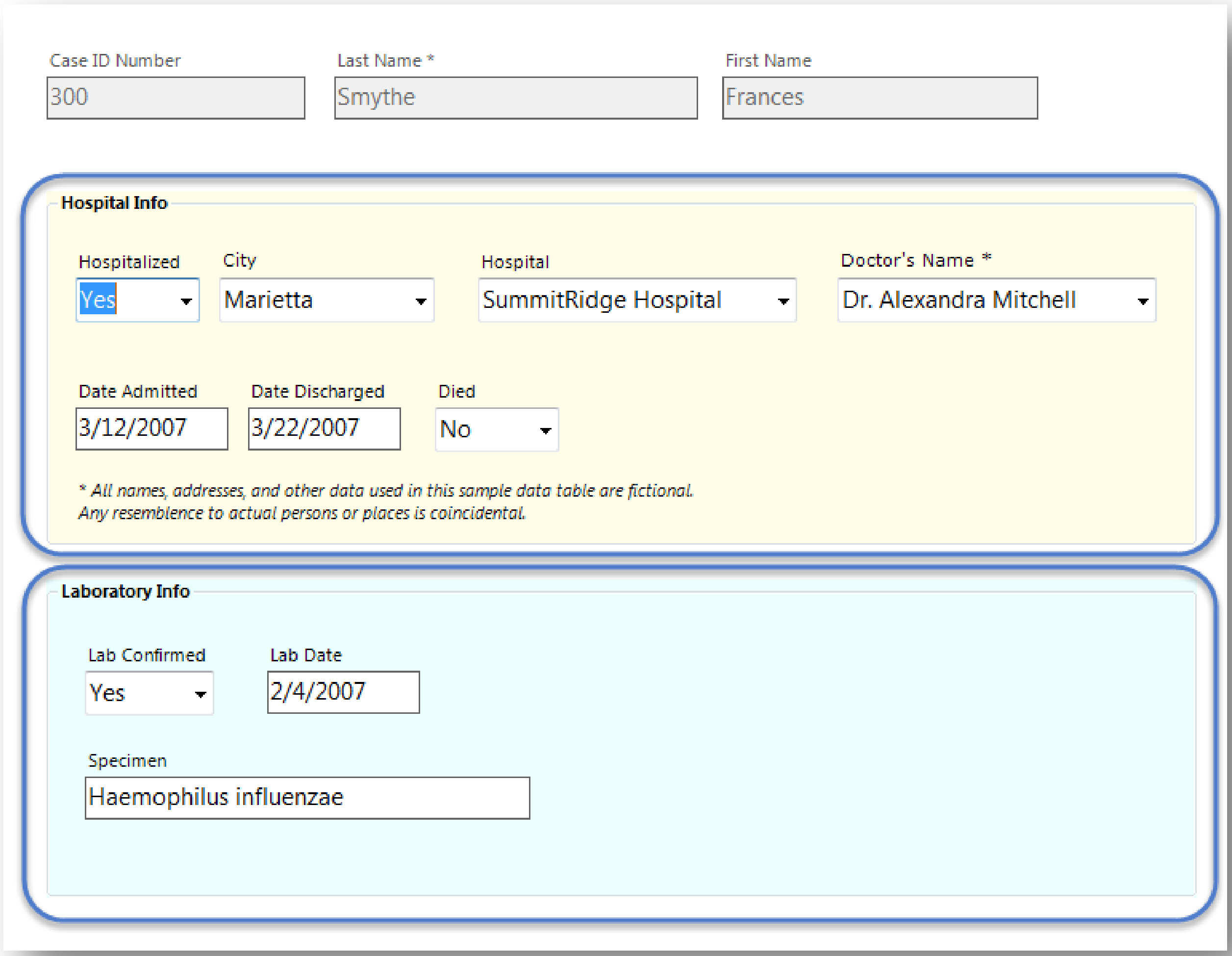
A Group field is a way to group together one or more fields on a page for visual and practical purposes. Visually, a group helps to tie similar fields together into a common category. Grouped fields also provide organization and structure to the form. A background color can be added to a group to enhance its appearance. You can nest groups and arrange them so they overlap specific fields for situations where you need a Venn style “diagram” on your form.

Group fields also have several practical purposes. In Classic Analysis and the Visual Dashboard, you can run statistics on individual fields, but if the fields are contained within a group, then you can use the group field and the statistics will be done on each field within the group.

Similarly, several Check Code commands can be run on all fields within a group by using just the group name, rather than using each field name individually. This makes commands such as Enable, Disable, Clear, Highlight, Set-Required, Set-Not-Required, among others, much easier for you to write.

Fields contained within the Group box are automatically members of the Group field. To remove a field from a group, simply drag the field outside of the boundaries of the group box. Likewise, dragging the group box itself will move all the fields within the group. When the group box is released, any additional fields within the boundaries of the group box are automatically members of the group field for analyses and Check Code execution.

The following figure provides an example of how Group field (circled in blue) appear in Enter.



Case ID Number	Last Name *	First Name
300	Smythe	Frances

Hospital Info

Hospitalized	City	Hospital	Doctor's Name *
Yes	Marietta	SummitRidge Hospital	Dr. Alexandra Mitchell

Date Admitted	Date Discharged	Died
3/12/2007	3/22/2007	No

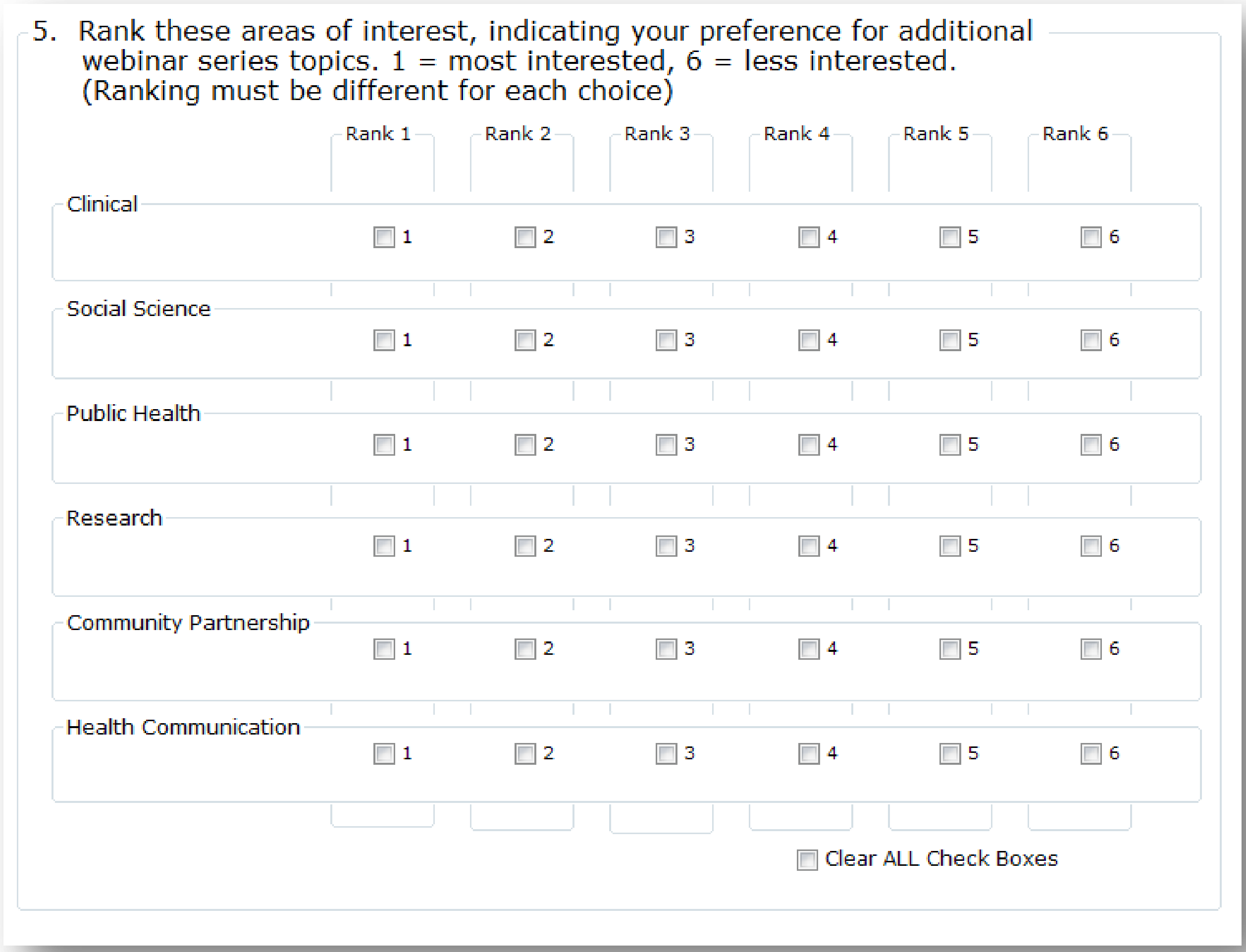
** All names, addresses, and other data used in this sample data table are fictional. Any resemblence to actual persons or places is coincidental.*

Laboratory Info

Lab Confirmed	Lab Date
Yes	2/4/2007

Specimen: Haemophilus influenzae

Figure 7.79: Group field



5. Rank these areas of interest, indicating your preference for additional webinar series topics. 1 = most interested, 6 = less interested. (Ranking must be different for each choice)

	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 6
Clinical	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5	☐ 6
Social Science	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5	☐ 6
Public Health	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5	☐ 6
Research	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5	☐ 6
Community Partnership	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5	☐ 6
Health Communication	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5	☐ 6

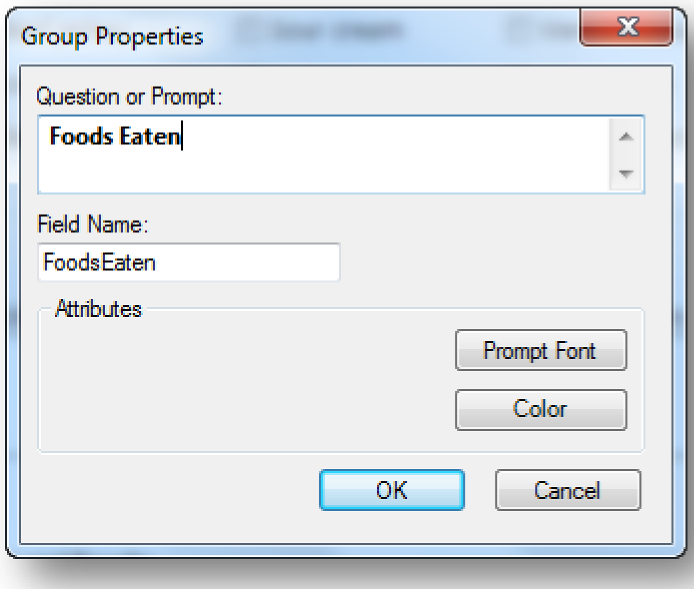
☐ Clear ALL Check Boxes

Figure 7.80: Nested Venn-style Group fields

To create a Group field:

1. Click and drag a **rectangle** around the fields to be grouped. Each field in the selection appears highlighted with a rectangle.

2. From the Form Designer toolbar, select **Insert > Group** or right click on a blank part of the canvas and select **New Group Field** from context menu. The **Group Properties** dialog box appears.



Question or Prompt: Foods Eaten

Field Name: FoodsEaten

Figure 7.81: Group Field Properties

3. Enter the **Question** or **Prompt**. The text entered in this field will display on the canvas and prompt the user to enter a response.

4. Click in the **Field Name** text box or press the tab key. Epi Info automatically suggests a field name based on the Question or Prompt, however, it is very important that field names be short, intuitive, and usable. The field name is used for data validation in Check Code and when doing analyses. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

   Note: It is best to simplify the field name at this time. Field names cannot be changed after data collection starts.

5. Click **OK** to create the group. The fields appear in the group box.

   - Move the group by clicking and holding the **group name** with the mouse.
   - Resize the group box by moving the cursor over the group and adjust the box dimensions by clicking and dragging the corner boxes to the desired dimensions. If the new size includes additional fields, they become members of the group.
   - Fields in the group are rearranged in the same process as on the Form Designer canvas. Click on the field and drag the field to the desired location.

Edit an Existing Field

Right click on the field and select **Properties** from the context menu. The Field Definition dialog box appears. Make the necessary changes and click **OK**.

Edit a Field Name

Field Names are critically important to Epi Info™ 7 projects because they are the basis for everything from database management, to data validation, and analysis. Because field names are such an integral part of Epi Info™ 7 projects**, field names cannot be changed after the data table has been created** to begin collecting data—even when testing the form with test data.

Therefore, it is very important to establish a concise and usable naming convention for your field names when beginning to design your data entry form. Although Epi Info automatically suggests a field name based on the Question or Prompt, it does so by concatenating the first several words together, stripping spaces and invalid symbols. This can be convenient when the question is only a few short words, but if the prompts to your questions frequently begin with similar phrases such as “Please enter the …?” or “What is the …?”, then the suggested field names will also have the same ambiguous start creating a very challenging and tedious experience searching for a specific field when writing Check Code or using Analysis and the Visual Dashboard.

For these reasons, is very important that field names be short, intuitive, and usable. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character “_” is permitted).

The field name cannot be changed after the data table is created. When testing a form in Enter, the data table is initially created. If the data table has no data or only test data that can be discarded, you can delete the data table in order to change the field name. Refer to the topic **‘Delete an Existing Data Table Without Deleting the Form’** for more information.

Invalid Field Names

Since field names cannot start with a number, if your question or prompt begins with a number, Epi Info’s suggested field name will begin with a leading letter “N” before the number.

Field names cannot have spaces or non-alphanumeric characters other than the underscore.

Since fields are referenced by their field name in the database, the field name must be unique—it cannot already exist as the name of another field in your form.

Finally, Epi Info needs to reserve some words for internal use and to avoid conflicts related to database interaction. For a complete list of reserved words, please see the appendix

If your desired field name is invalid for any of the reasons mentioned above, it will appear in red font and the OK button will be disabled. To avoid this, modify the field name slightly, perhaps add a suffix or prefix to make it an acceptable name. Then click **OK**.

Delete a Field

1. Right click on the **field**. The context menu opens.
2. Click **Delete**. The field is removed from the form.

Warning: The field and any data previously collected are deleted from the form and database. Deletions occur immediately. There is no prompt to verify the deletion before it occurs, and the only way to recover the field is by using the "undo" feature, however, "undo" will not recover deleted data.

Copy, Cut, and Paste Fields

1. Click and drag a **rectangle** around the fields to be copied or cut.
2. From the Form Designer toolbar, select **Edit > Copy** or **Cut** from the drop-down list.
3. Click in the **new section** of the form or select a **new page** in the project.
4. Select **Edit > Paste**. The copied fields appear in the form.
   When pasting fields, Epi Info automatically suggests a field name based on the name of the original field. To ensure the field name is unique, Epi Info appends a number. For example, if the field name of a field being copied is "Age", then the pasted field is automatically named "Age1". Copying and pasting this field again will result in a third field named "Age11". The similarity of these names may lead to confusion and sometimes errors when used in Check Code and in analyses. Therefore it is important to change the field name as described in the next step.
5. For each field pasted, do the following:
   1. Right click the new field's question or prompt and select Properties from the context menu.
   2. Click in the Field Name text. Change the field name making it more clear and intuitive for the instance of the new field.
   3. Field names cannot start with a number or contain any spaces or non-alphanumeric characters (except the underscore character "_" is permitted).

Note: Field names cannot be changed after a data table has been created (data collection has started).

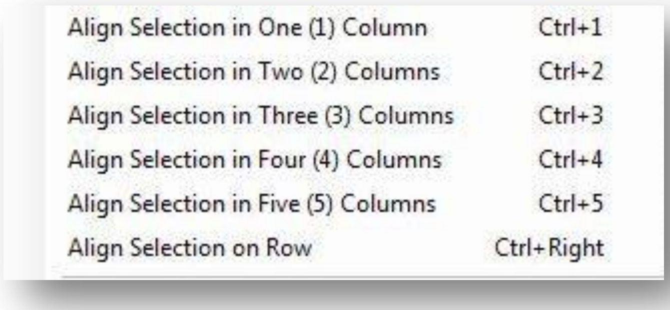
Note: ***You can also right click on the field and use the context menu to copy or cut the field. Also, you can right click the canvas and use the context menu to paste a field from the clipboard.***

Align Fields

There are several ways to align fields in Form Designer. Aligning fields in a consistent format allows you to make the form appear more professional. The process of aligning fields begins with selecting the desired fields to be aligned.

1. Click and drag a **rectangle** around the fields to be aligned.
   - To align the fields vertically, select **Format > Alignment > As Stack.**
   - To align the fields horizontally, select **Format > Alignment > As Table.**

- For additional options, right click on the canvas to display the possible alignment configurations.



Align Selection in One (1) Column	Ctrl+1
Align Selection in Two (2) Columns	Ctrl+2
Align Selection in Three (3) Columns	Ctrl+3
Align Selection in Four (4) Columns	Ctrl+4
Align Selection in Five (5) Columns	Ctrl+5
Align Selection on Row	Ctrl+Right

Figure 7.82: Field Alignment Options

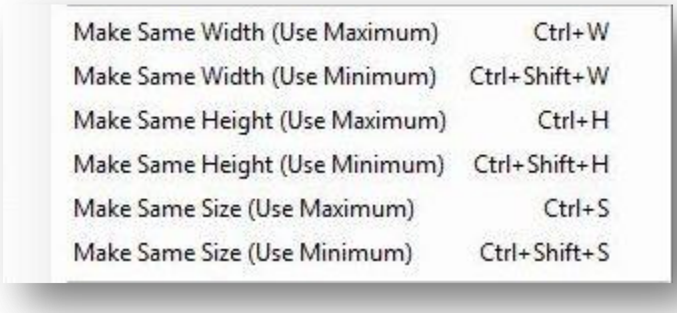
2. The options appear as follows:

   - **Align Selection in Column(s)** – aligns all fields vertically in the number of columns indicated (similar to **Format > Alignment > As Stack**).
   - **Align Selection on Row** – aligns all fields horizontally (similar to **Format > Alignment > As Table).**

Set Field Size

There are several field sizing options available in Form Designer. Making field sizes consistent will allow you to make the form appear more professional. The process of standardizing the field size begins with selecting the desired fields.

1. Click and drag a **rectangle** around the fields to be re-sized.
2. Right click on the canvas to display the possible sizing configurations.



Make Same Width (Use Maximum)	Ctrl+W
Make Same Width (Use Minimum)	Ctrl+Shift+W
Make Same Height (Use Maximum)	Ctrl+H
Make Same Height (Use Minimum)	Ctrl+Shift+H
Make Same Size (Use Maximum)	Ctrl+S
Make Same Size (Use Minimum)	Ctrl+Shift+S

Figure 7.83: Adjust Field Size

3. The options appear as follows:

   - **Make Same Width (Use Maximum)** – Adjusts selected fields' width to match the widest field of the selection

- **Make Same Width (Use Minimum)** – Adjusts selected fields' width to match the narrowest field of the selection
- **Make Same Height (Use Maximum)** – Adjusts selected fields' height to match the tallest field of the selection
- **Make Same Height (Use Minimum)** – Adjusts selected fields' height to match the shortest field of the selection
- **Make Same Size (Use Maximum)** – Adjusts selected fields' height and width to match the largest field of the selection
- **Make Same Size (Use Minimum)** – Adjusts selected fields' height and width to match the smallest field of the selection

Manually Set Field Size

1. Hover your mouse over the field to display the field's frame and sizing handles.
2. Hover the mouse over one of the sizing handles to see the mouse change to an arrow cursor.
3. Click and drag a sizing handle to stretch the field to the needed size.

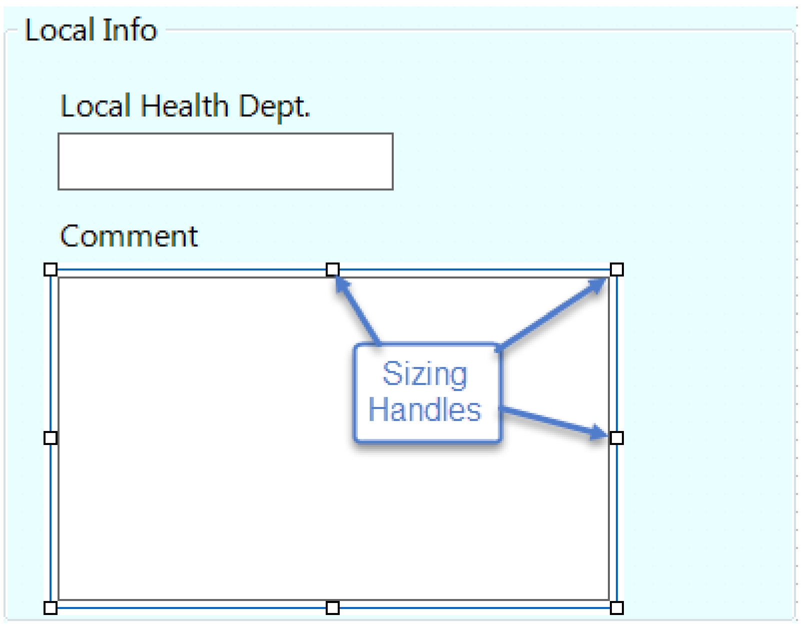


Figure 7.84: Manually Adjust Field Size

Manually Set Field Position

In certain cases, the field alignment or positioning will need to be adjusted manually. In Epi Info, the field and prompt can move together or separately. This is especially helpful if using a right-to-left language where the question or prompt should be to the right of the entry field.

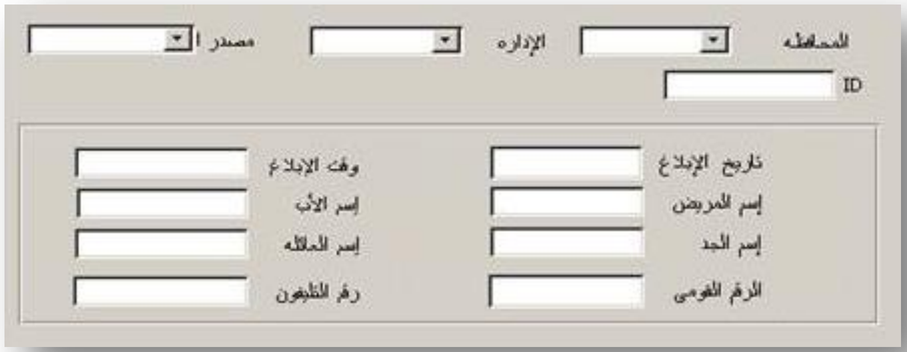


Figure 7.85: Manually Adjust Field Position

The following example is based on a form containing a Label/Title field and a Text field.

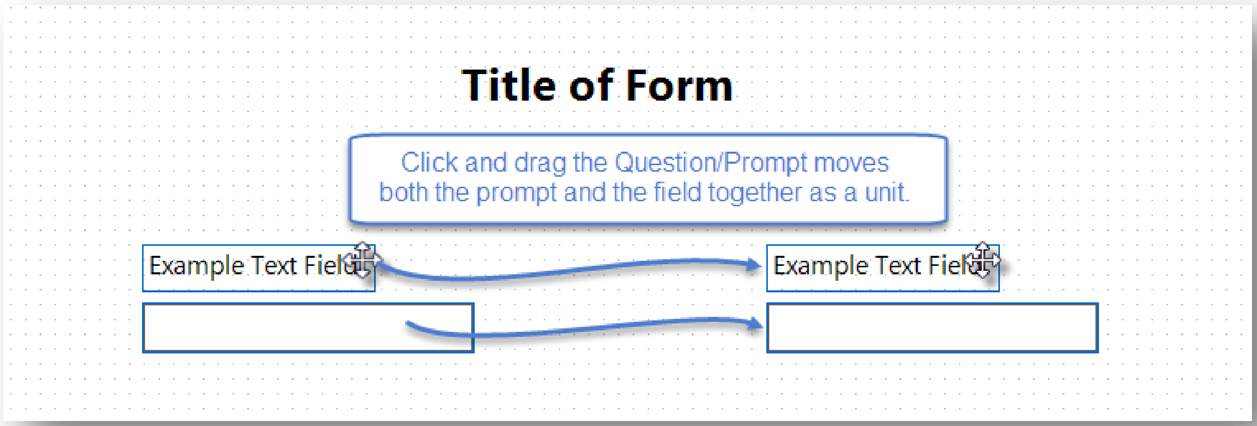

Title of Form
Click and drag the Question/Prompt moves both the prompt and the field together as a unit.
Example Text Field
Example Text Field

Figure 7.86: Prompt and Entry Field Selection

1. To move a field and question together, click and drag the field's prompt.
2. To move the field and prompt separately, click and drag the entry field, not the prompt, as shown below.

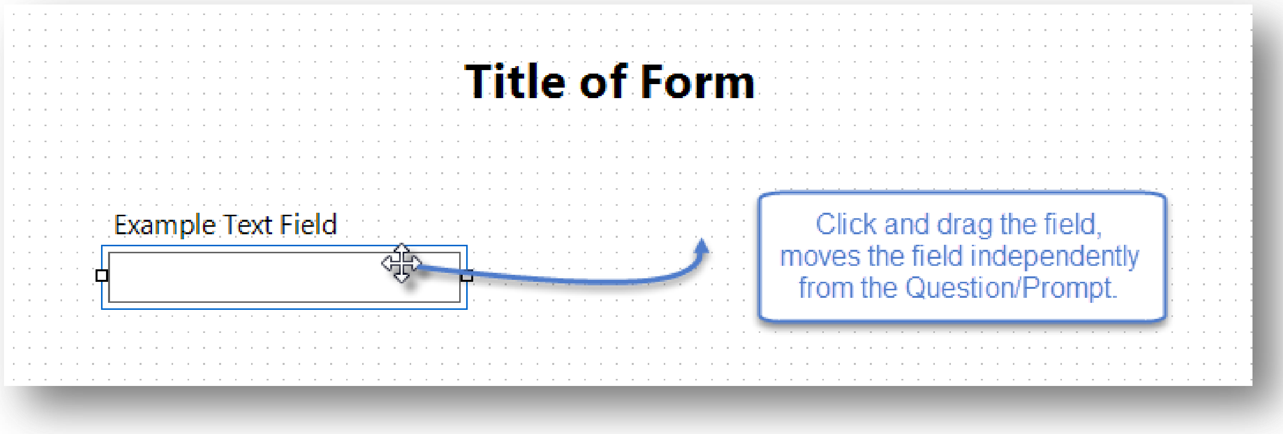

Title of Form
Example Text Field
Click and drag the field, moves the field independently from the Question/Prompt.

Figure 7.87: Select Entry Field

3. **Release** the entry field at the desired location.

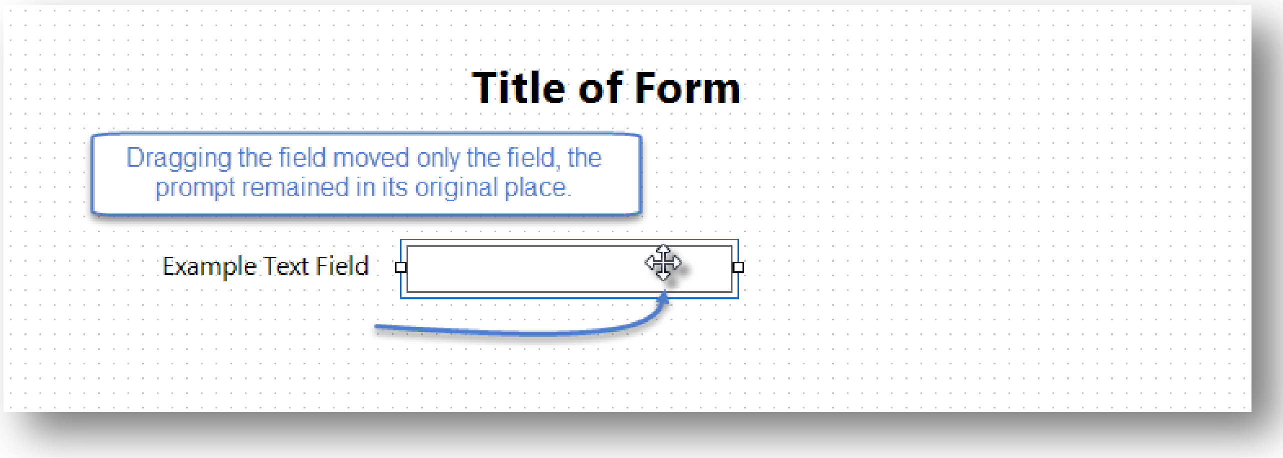

Title of Form
Dragging the field moved only the field, the prompt remained in its original place.
Example Text Field

Figure 7.88: Adjust Entry Field Location

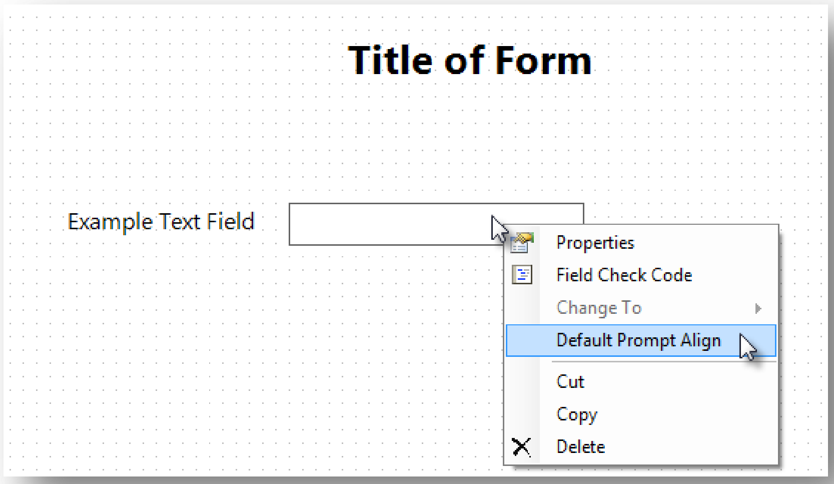


Figure 7.89: Default Prompt Alignment

4. To restore the original field and prompt alignment according to the default configuration, right click on the **field** or prompt and select **Default Prompt Align**.

5. To set the default prompt alignment, select **Format > Page Setup** from the Form Designer menu bar. Set the default alignment to be vertical so the field is below the prompt and left aligned, or horizontal where the field is to the right of the prompt on the same row.

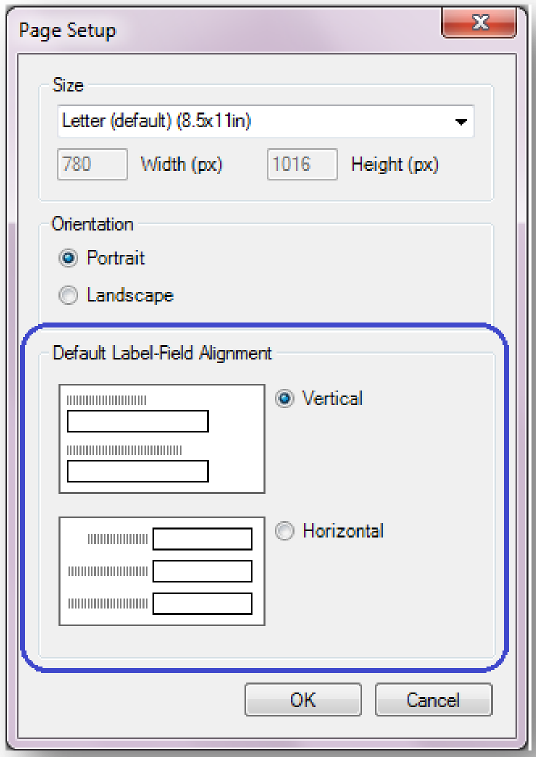


Figure 7.90: Page Setup dialog showing Default Label-Field Alignment

Tab Order

In the Enter module, during data entry, pressing tab or enter will move the cursor to the next field. Tab Order is the order or sequence that the cursor moves from field to field. Initially, the tab order is determined by the order in which the fields are added to the form. In many cases, as fields are moved around the page, added from a template, or pasted from somewhere else, the tab sequence may become disorganized. There are other situations in which you may prefer the tab order to move in a custom way, such as down a column of questions, or maybe from right to left across the page.

The proper tab order will make entering data more user friendly and efficient. Additionally, the tab order will determine how fields are displayed on a mobile device.

To show the tab order, right click on the canvas and select **Tabs > Show Tab Order**. The tab order number for each field is shown in a rectangle near the entry field. Red rectangles indicate the field is a Read Only field or one that cannot receive the cursor, such as

Label/Title fields. Group fields also do not receive the cursor, but a tab number is reserved for the group, even though the tab order number for a Group box is not shown.

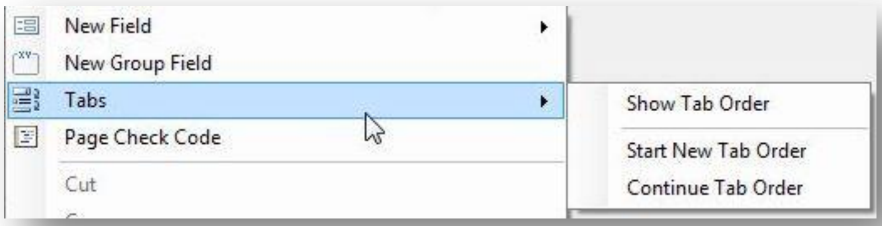


Figure 7.91: Show Tab Order

Start New Tab Order

Right click the canvas and select **Tabs > Start New Tab Order** from the context menu. Selecting this function will automatically set the tab order according to each field's position from left to right, then top to bottom.

Customize the Tab Order

You can customize the tab order to meet the needs of your questionnaire. This process begins by showing the tab order on the canvas. The example below demonstrates how to adjust the tab order in the Food History form of the EColi project located in the Projects folder.

Right click the canvas and select **Tabs > Show Tab Order** from the context menu to show the current order of field entry.

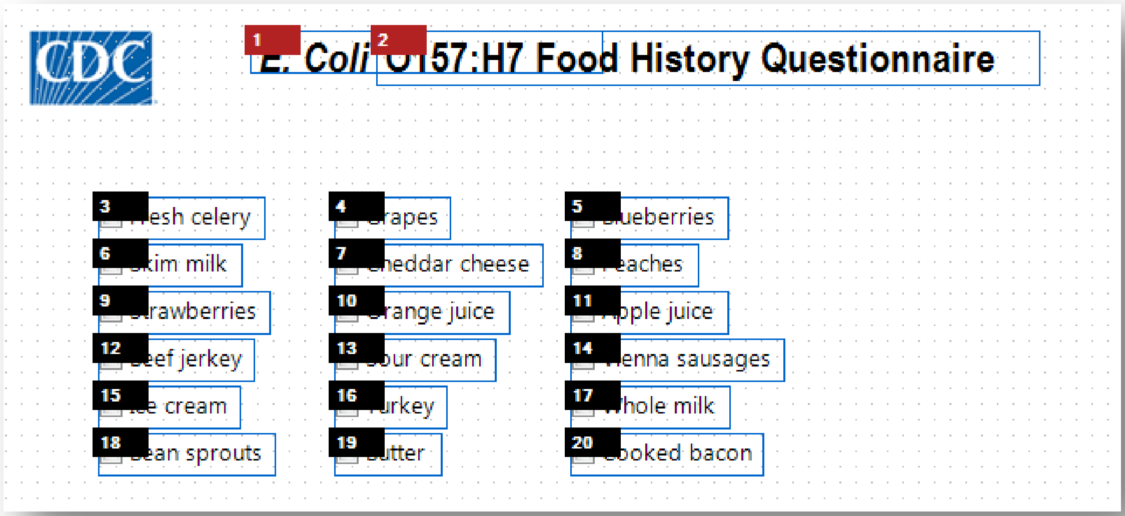


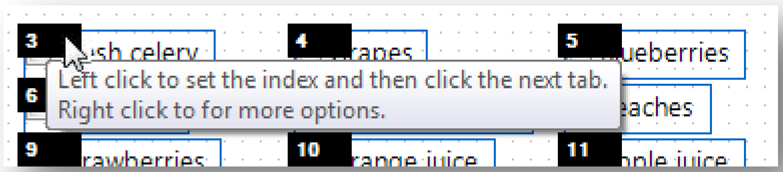
Figure 7.92: Show Tab Order

In the example shown in the figure above, the sequence will move the cursor from left to right, row by row.

Suppose you wanted to have the order run down the first column, from 'Fresh celery' to 'Bean sprouts', then over to the 'Grapes' in the second column. There are a couple of methods to do this.

Customize Tab Order using the Mouse

1. With the Tab Order showing, hover the mouse pointer over the field you want to be the first field in the sequence. In this example, we want to start the sequence with 'Fresh celery' at tab order 3. A prompt appears showing the options available to reset the tab order.



Left click to set the index and then click the next tab.
Right click to for more options.

Figure 7.93: Tab Order Prompt

2. Click on the **tab order number** to set the index. In this example, we click the number **3** in order to set the index at 3.
3. Click on the tab order number for the next field you want to follow the index field. In our example, the field below 'Fresh celery' is 'Skim milk', currently tab order 6. When we click on the number 6, the number changes to follow our index field and becomes 4 (one more than 3—our index).

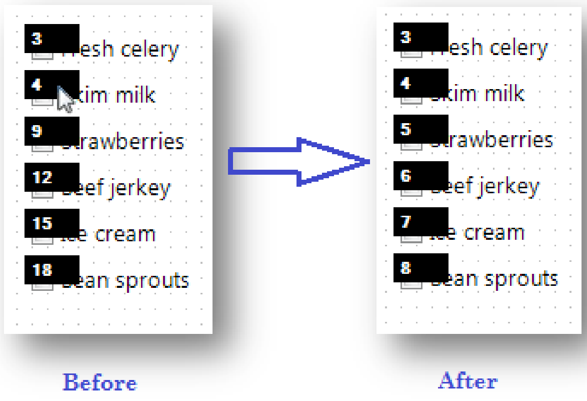


Figure 7.94: Tab Order Adjusted Manually – Before and After

4. Click each successive field's **tab order** box in the desired flow and the numbers are updated sequentially. In our example, we clicked on the tab order boxes 9, 12, 15, and 18 to set the order to be 5 through 8 consecutively.

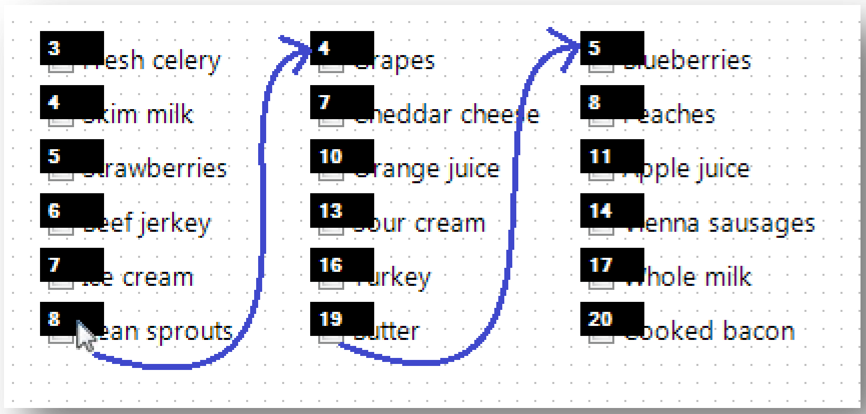


Figure 7.95: Additional Tab Order Options

5. After clicking the tab number at the bottom of the first column, move the mouse up to the **first field** of the second column to continue the sequence. Continue clicking each successive **tab order number** until you are finished. Click anywhere on the canvas to hide the tab order numbers.

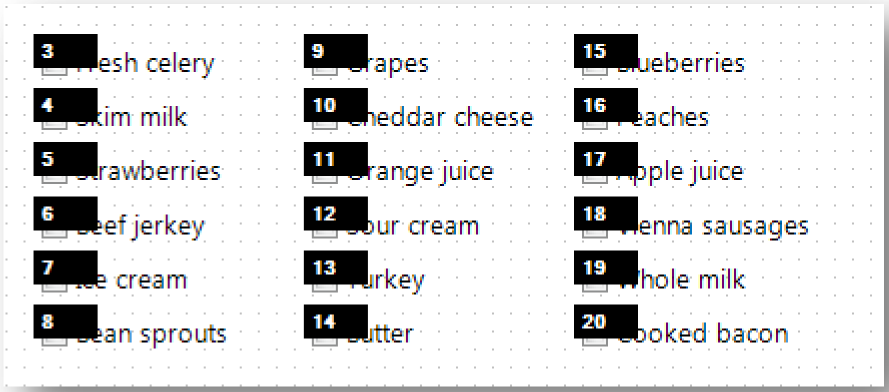


Figure 7.96: Additional Tab Order Options

Customize Tab Order using 'Continue Tab Order'

The Continue Tab Order feature sequentially renumbers the tab order for a selected set of fields based on the current index value. As an example, the figure below shows the original tab order in the Food History form of the EColi project located in the Projects folder.

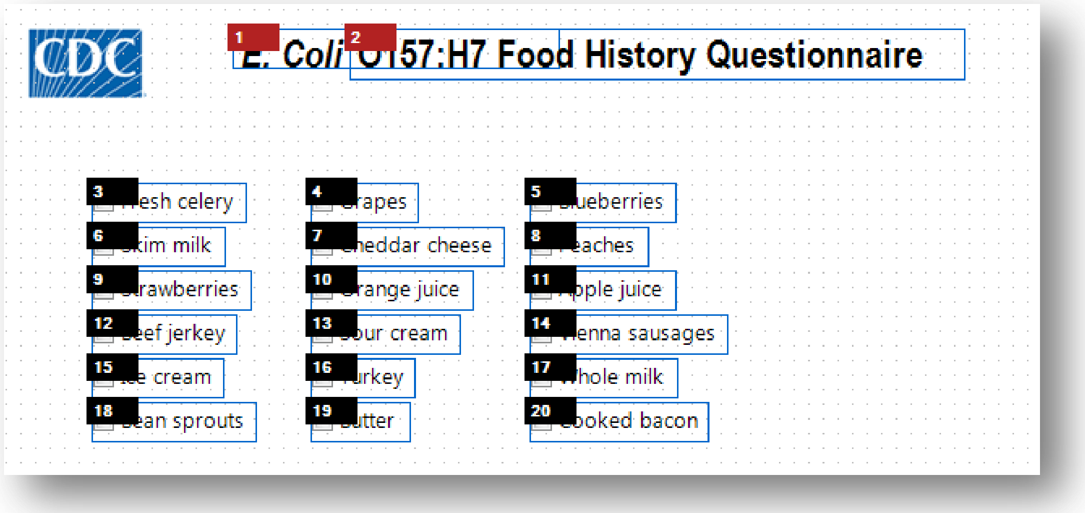


Figure 7.97: Customize Tab Order

In order to have the order run down the first column, from 'Fresh celery' to 'Bean sprouts', then over to 'Grapes' in the second column using the Continue Tab Order option, follow these steps:

1. Click and drag a rectangle around the first column of fields. This will select just those fields.

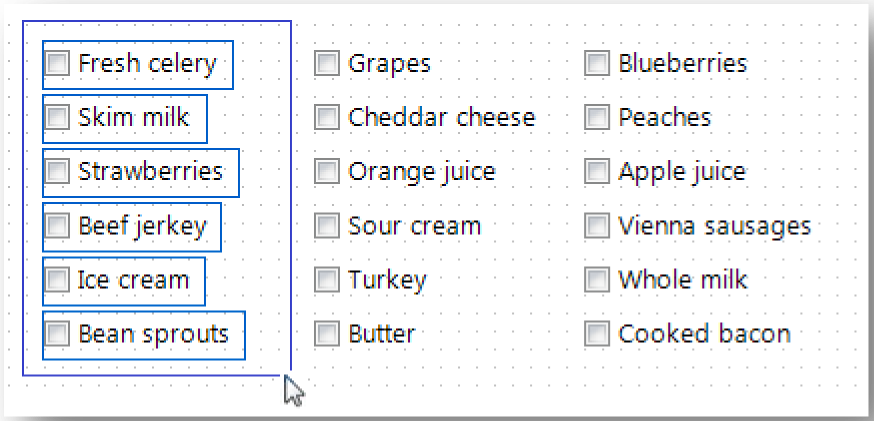


Figure 7.98: Click and Drag Column of Fields for Tab Order

2. Right click the canvas and select **Tabs > Start New Tab Order**. This will reorder only the selected fields beginning with the first tab order number.

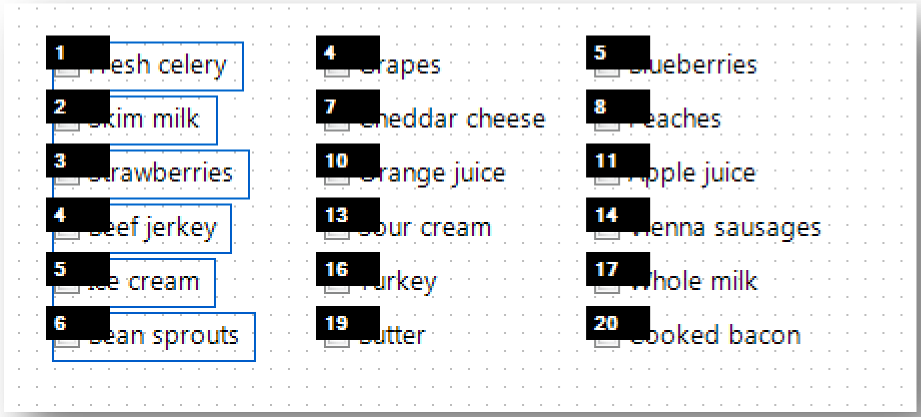


Figure 7.99: Start New Tab Order

3. **Click** and **drag** a **rectangle** around the second column of fields to select them.

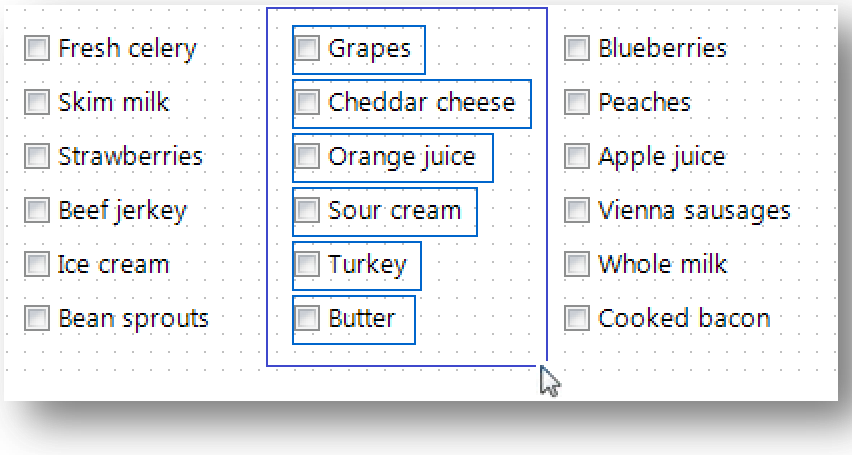


Figure 7.100: Grouping Tab Order

4. Right click on the canvas and select **Tabs > Continue Tab Order**. This will reorder the second column of fields continuing from the last Tab Order Number used in the previous action.

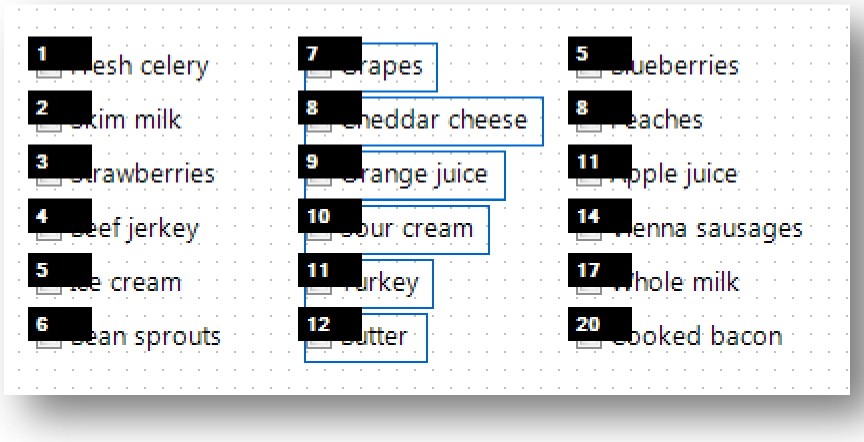


Figure 7.101: Continue Tab Order

5. Repeat steps 3 and 4 until all fields' Tab Order are renumbered as needed.

Disable Tab

The Disable Tab feature makes the field essentially a Read-Only field that does not receive the cursor. In the example shown in the figures that follow, the Age field is calculated in Check Code based on the value entered for Birth Date. Therefore, the designer of this form wants the cursor to skip over the Age field by disabling its tab.

1. Right click the canvas and select **Tabs > Show Tab Order** from the context menu.

2. Right click on the desired field's tab order box and select Disable Tab.

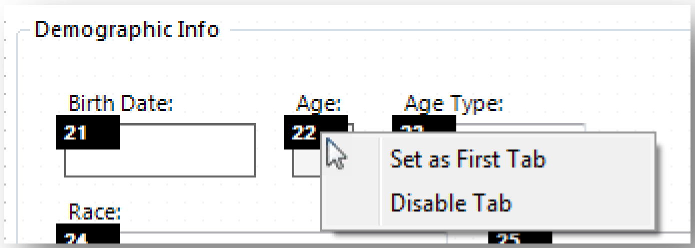


Figure 7.102: Disable Tab

After disabling the Tab for a field, the tab order box becomes red to indicate that it is no longer a field that will receive the cursor. Pressing the tab key will bypass any disabled tabs.

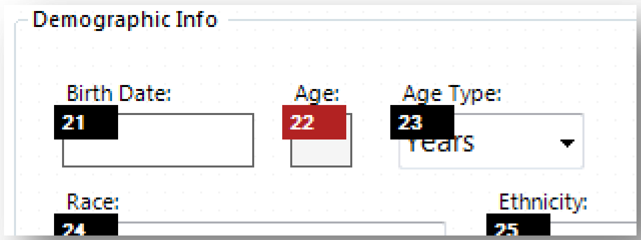


Figure 7.103: Bypassing Disabled Tabs

Note: Check Code only runs when the cursor enters a field. If a field's tab order is disabled, Check Code will not run for that field.

Templates

Templates are a time-saving feature that allows you to save all or part of your project to be reused in other projects and shared with other Epi Info™ 7 users *without* sharing data collected with the project. For example, you might want to save certain fields for use on forms you will create later, as opposed to having to create those same fields over and over (e.g. case demographic information). Saving fields for later use could also be a good way to have a library of fields already built and ready to go in case of a public health emergency. Templates are saved to the Epi Info™ Templates folder using the Extensible Markup Language (XML) format. There are four types of templates: Field, Page, Form, and Project.

Epi Info™ 7 is packaged with a number of project and field level templates that demonstrate many of the various field types and features of Epi Info™ 7 including some Check Code examples. These are listed in the Project Explorer under the Templates category.

One of the best ways to see how Epi Info can be used is to create a project using these templates and examine how they were done. A new project can be created from one of the sample Project Templates in the Form Designer when you open it for the first time.

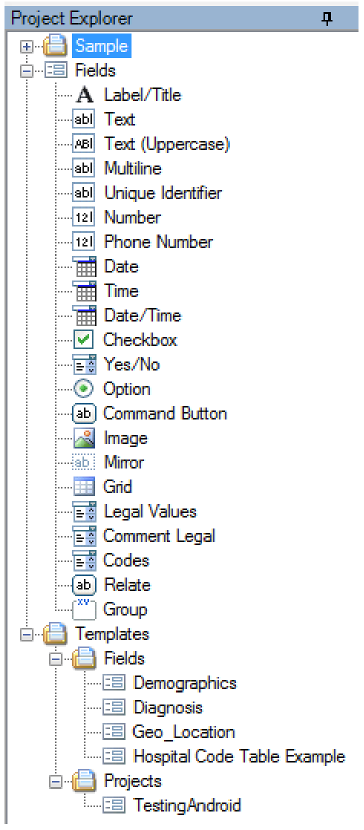


Figure 7.104: Project Explorer Template

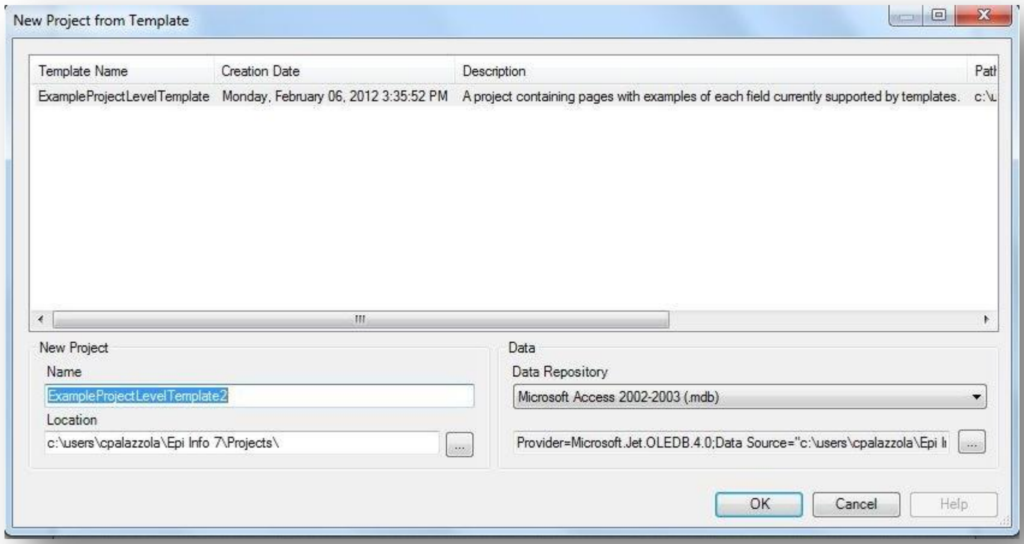
Project Templates

Since Epi Info™ 7 Projects contain one or more forms, and each form has one or more pages with fields, Project Templates contain an example of all of the above including any Check Code that may have been saved with the template. Project templates are an effective and convenient way to share projects between colleagues without including collected data.

Create a New Project From Template

A list of demonstration templates are located in the Project Explorer under **Templates > Projects.** Use one of them to create a new project.

1. **Double-click** the template of choice or select **File > New Project from Template.** The New Project from Template dialog opens.



New Project from Template

Template Name	Creation Date	Description	Path
ExampleProjectLevelTemplate	Monday, February 06, 2012 3:35:52 PM	A project containing pages with examples of each field currently supported by templates.	c:\u

New Project
Name: ExampleProjectLevelTemplate2
Location: c:\users\cpalazzola\Epi Info 7\Projects\

Data
Data Repository: Microsoft Access 2002-2003 (.mdb)
Provider=Microsoft.Jet.OLEDB.4.0;Data Source="c:\users\cpalazzola\Epi I

Figure 7.103: New Project from Template

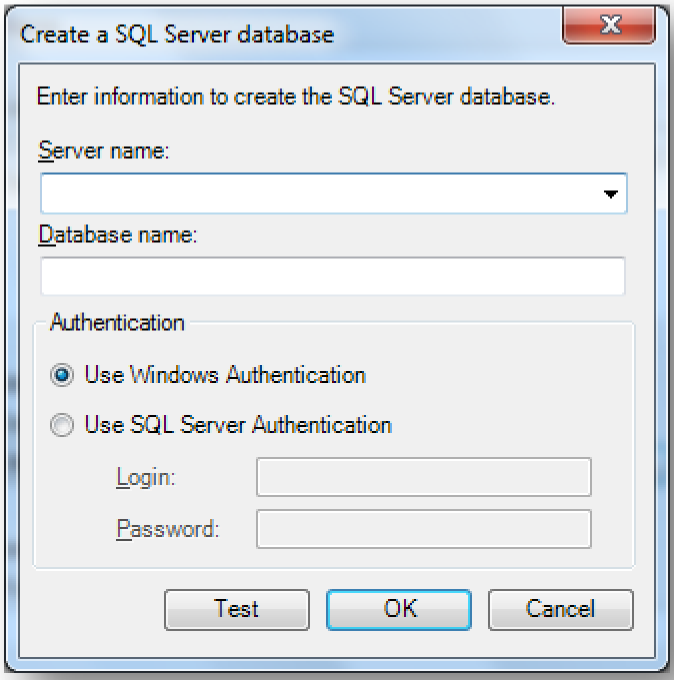
2. Select a template from the list. The Name and Data Repository used when the template was created is automatically populated in their respective fields.

3. **Edit** the **Project Name**, if necessary (optional).

4. The default location for the project is the Projects folder within the Epi Info™ 7 folder. Specify another location, if necessary (optional).

5. Select the database format from the **Data Repository** drop-down list. The default option is Microsoft Access 2000-2003, however SQL Server is also available. To use the SQL Server option, you need to have access to a SQL Server database.

6. If you selected Microsoft SQL Server for the Data Repository, then click the browse button (…) to the right of the Data Repository field to enter the connection information for the SQL Server database. Contact your SQL Server database administrator for the required information requested in this dialog.



Enter information to create the SQL Server database.

Server name:

Database name:

Authentication

[✓ Use Windows Authentication]

Use SQL Server Authentication

Login:

Password:

Figure 7.1054: SQL Server Database Dialog Box

7. Click **OK**. The Form Designer creates the project and after a few moments, the new project appears with the first form's initial page shown on the canvas.

For additional information on creating new projects, refer to the topic Create a New Blank Project and Form.

Create a New Template from a Project

There are two options to create a project template that includes all the forms, pages, fields, and Check Code in the current project – **Save Project as Template**, which allows you to specify a template name and description, and **Quick Save Project as Template** which automatically names the template based on the project name and the current date and time. Both options save the template in the same way, but Quick Save does not allow for a custom name and description.

1. In the Project Explorer, right click on the **project name**. The project name is the top most item in the Project Explorer.

2. From the drop-down list, select **Save Project as Template** or **Quick Save Project as Template**. If Quick Save Project as Template is selected, the template is immediately saved to the Templates > Projects folder with the name of the project and a timestamp.

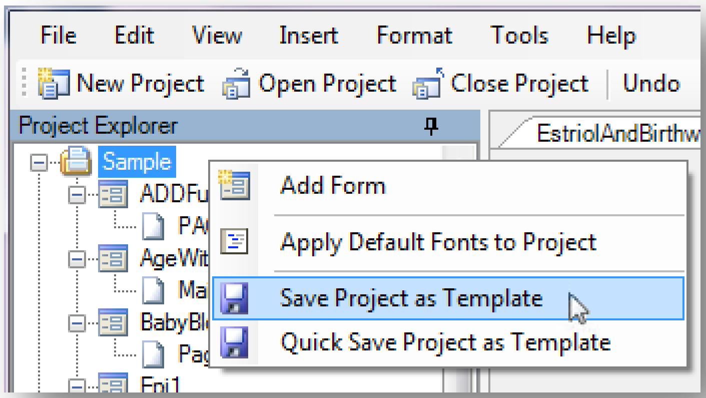
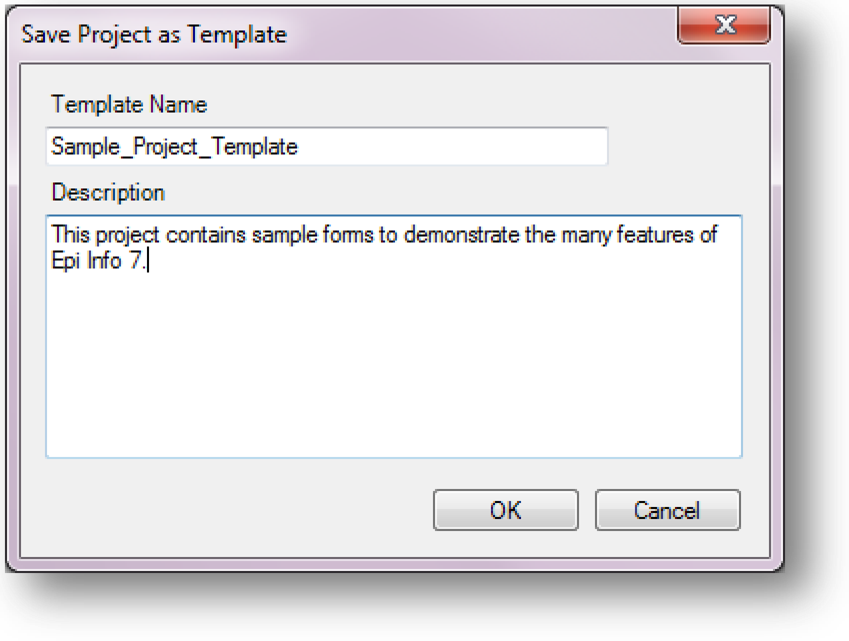


Figure 7.1065: Save Project as Template

3. If the **'Save Project as Template'** option is selected, a dialog appears to allow you to add a Template Name and an optional Description.



Sample_Project_Template

This project contains sample forms to demonstrate the many features of Epi Info 7.

Figure 7.1076: Save Project as Template Description

4. Click **OK**. The template is saved to the **Templates > Projects** folder.

5. To retrieve the template to send to someone, locate the template in the Project Explorer.

6. Right click the template name and select **Open Containing Folder**. This opens the folder in Windows Explorer. From Windows Explorer, you can copy and paste the XML template file into an email or save it to a shared network location for others to use.

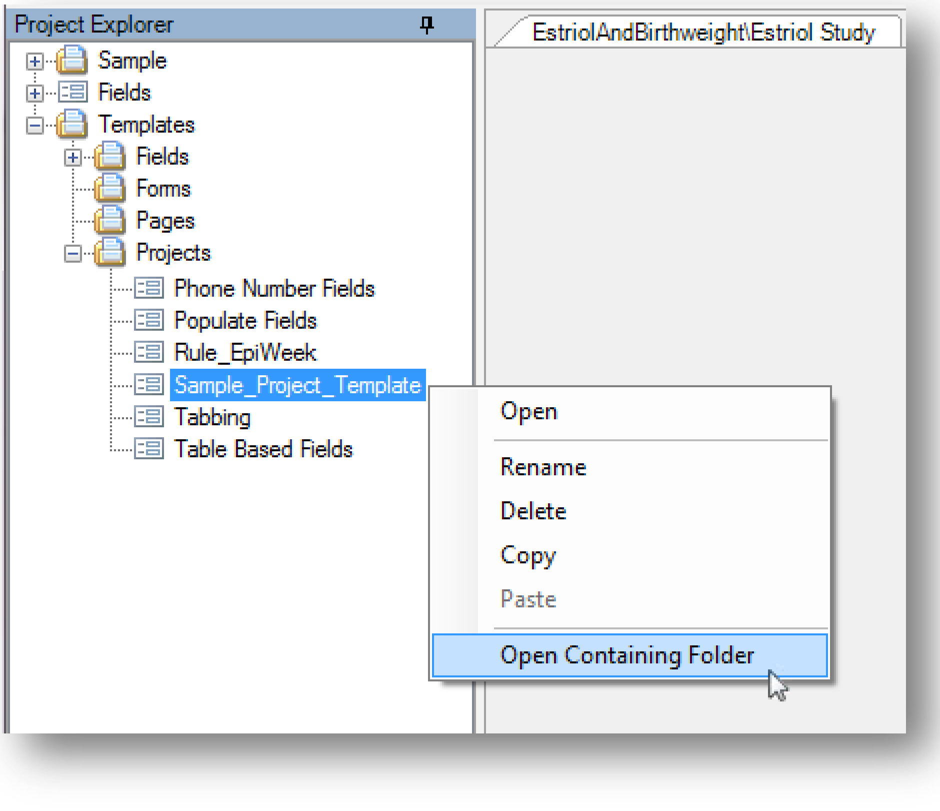


Figure 7.1087: Open Containing Folder

Get Template

If you receive a template from a colleague, an easy way to copy the template to the correct location within Epi Info™ 7 is to use the Get Template feature.

1. From the Form Designer menu, select **File > Get Template…** The Open dialog appears.
2. Locate the **template file**.
3. Click **Open**. The template is saved to the corresponding **Epi Info 7 > Templates** folder according to the type of template. If the template is a Project template, then the New Project from Template dialog opens.

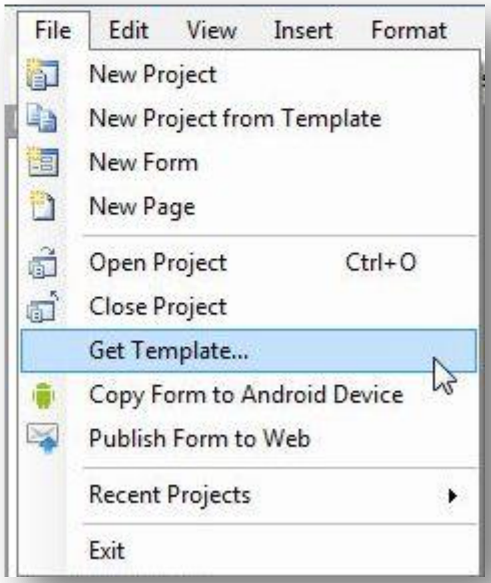


Figure 7.1098: Get Templates

Form and Page Templates

Form and Page templates are created and used in similar ways. The difference is in the item selected when right clicking to get the context menu. Also, form templates can be used to create a single form project, whereas a page template can only be added to a form in an exising project.

Create a Form Template

If you want the template to include all the pages and associated Check Code for a given form then right click the form name in the Project Explorer and select Save Form as Template. In the example shown below, the form 'Surveillance' along with its three pages, 'Person', 'Hospital Info', and 'Case Report', will be included in the template along with all associated Check Code.

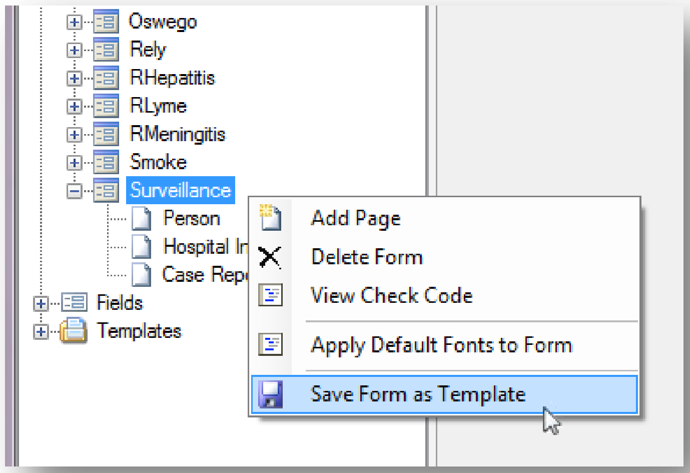


Figure 7.1109: Save Form as Template

Create a Page Template

If you want the template to include the contents of a specific page, and the Check Code for the fields on that page, then right click the page name in the Project Explorer and select Save Page as Template. In this example, only the selected page 'Hospital Info' and its Check Code will be in the template.

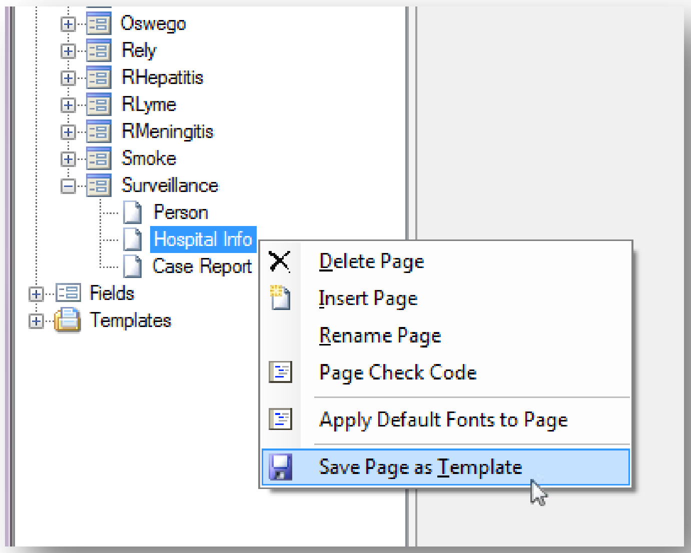


Figure 7.111: Save Page as Template

The **Add Template** dialog appears for you to enter a **Template Name**.

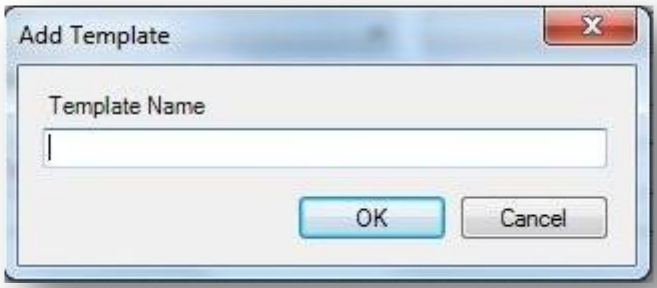


Figure 7.112: Add Template dialog box

Click **OK**. The page template is created and saved to the respective folder under **Templates** in the **Project Explorer.**

How to Use Form and Page Templates

Form and Page Templates are very similar in the way they are used. In order to use either template, there must be an existing Project already open in the Form Designer. The difference is that a Form Template contains the instructions to create a set of pages, each with one or more fields and Check Code, if any. A Page Template has only the page layout information and instructions to create the fields on a single page along with Check Code, if any.

Since all field names and page names must be unique within a given form, a new form created from a Form Template requires very little additional work to make the form usable after it is created from a Form Template.

A Page Template, on the other hand, can be used repeatedly on a given form. Because of this, during the process of creating the new page from a Page Template, if any field names already exist for the form, Epi Info™ 7 automatically appends a number to new field name from the template. Epi Info™ 7 does this to avoid having any duplicate field names. Unfortunately, this has the side effect of potentially breaking the Check Code that may exist for that page template. The field names within the Check Code are not synchronized. Therefore, the Check Code must be manually updated with the new field names.

To use a Form or Page Template, follow these steps:

1. With a project open in Form Designer, locate the template in the Project Explorer – **Templates > Forms** for form templates, or **Templates > Pages** for a page template.
2. If you cannot find the template you need, but you know it is saved somewhere, use the **File > Get Template…** feature to browse for the template.
3. Click and Drag the template to the canvas. Epi Info™ 7 will create the new form or page and place it in the Project Explorer.
4. If a page template was used, to change the page order, clicking and drag the page in the Project Explorer to the appropriate position.

Field Templates

Field templates allow for easy and concise duplication and sharing of fields and enable consistent collection of routine information. When a survey needs to have several questions asked in the same way, as in a **Likert scale**, it is easiest to set up one or two or a few of these questions, save them as a field template, then add additional sets of questions to the form from the template. This method only requires updating the prompt and field name properties of the new fields.

Each field template includes the field definition and attributes for one or more fields, along with any Check Code written for the selected fields. A list of demonstration field templates are available in the Project Explorer under **Templates > Fields.**

How to Create a Field Template

A convenient way to reuse a field or set of fields along with their Check Code or to share fields with a colleague, is to create a Field Template. To do this, follow these steps:

1. Select the fields to add to the template by drawing a rectangle around them. To do that, click and drag the mouse from the top left of the desired fields diagonally down to the lower right until all needed fields are contained in the blue box.

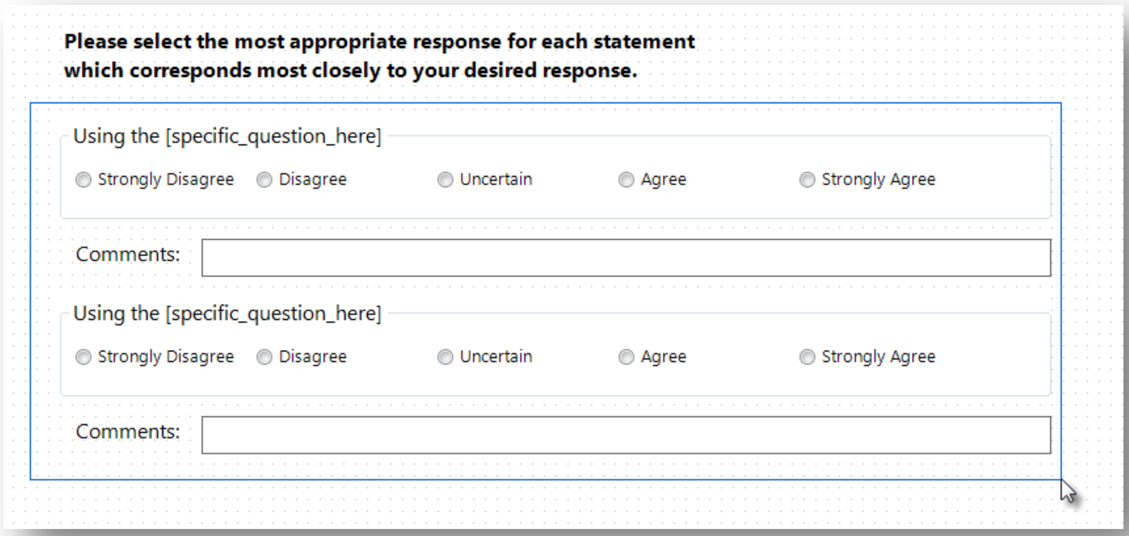

Please select the most appropriate response for each statement which corresponds most closely to your desired response.
Using the [specific_question_here]
Strongly Disagree
Disagree
Uncertain
Agree
Strongly Agree
Comments:
Using the [specific_question_here]
Strongly Disagree
Disagree
Uncertain
Agree
Strongly Agree
Comments:

Figure 7.113: Drag a rectangle to Select Fields

2. **Release** the mouse button and the selected fields will remain highlighted.

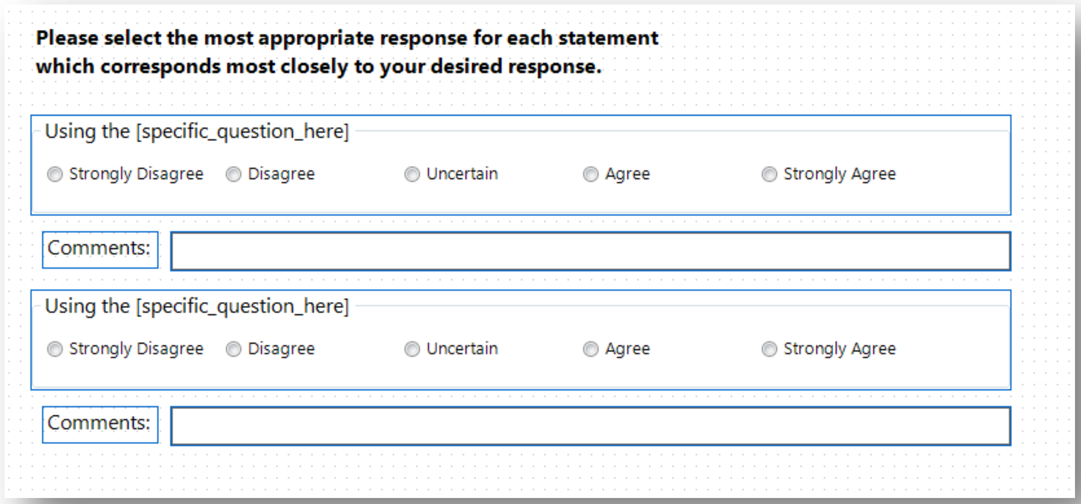

Please select the most appropriate response for each statement which corresponds most closely to your desired response.
Using the [specific_question_here]
Strongly Disagree
Disagree
Uncertain
Agree
Strongly Agree
Comments:
Using the [specific_question_here]
Strongly Disagree
Disagree
Uncertain
Agree
Strongly Agree
Comments:

Figure 7.114: Selected Fields appear with blue border

3. Right click on a blank area of the canvas. Select **Save Selection as Template** from the context menu.

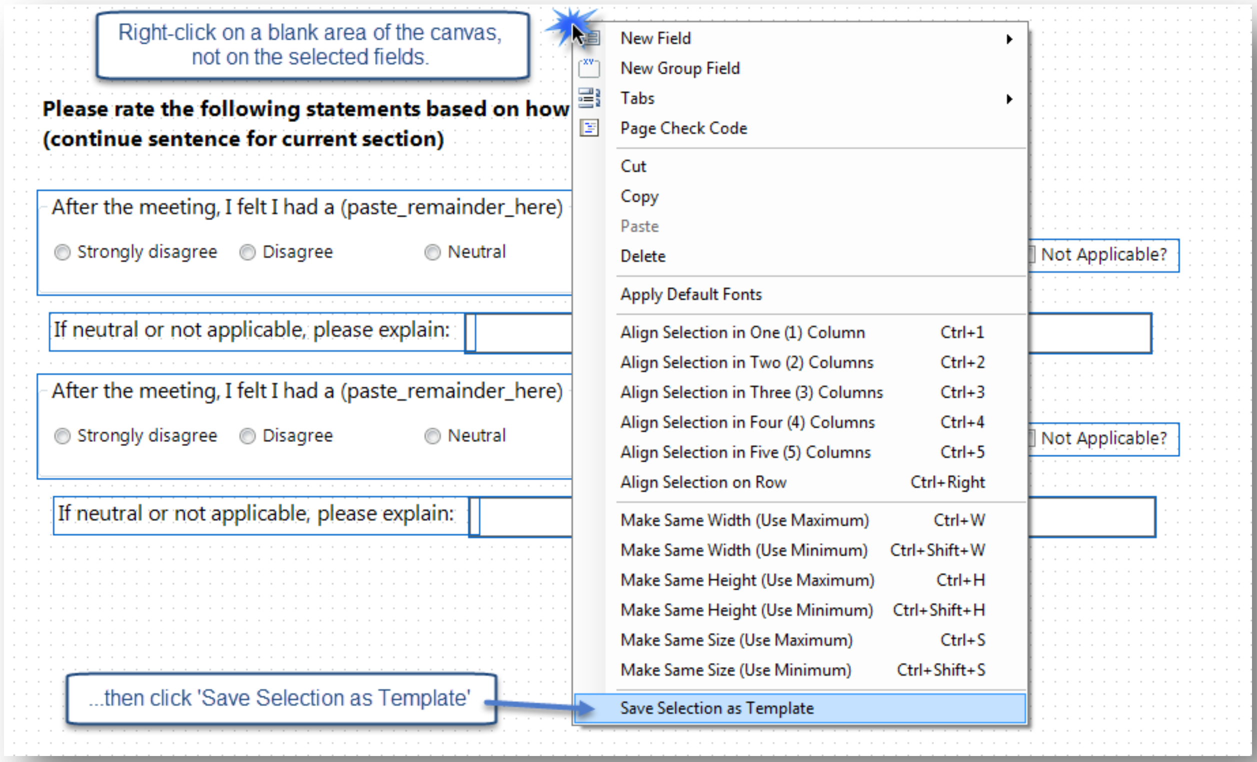
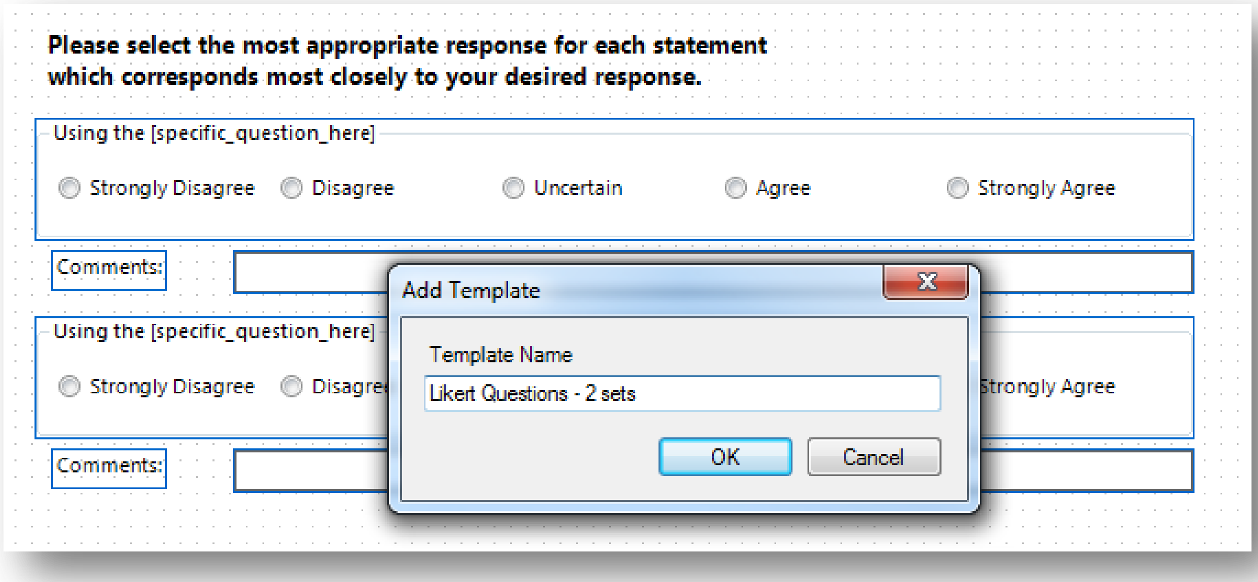

Right-click on a blank area of the canvas, not on the selected fields.
Please rate the following statements based on how
(continue sentence for current section)
After the meeting, I felt I had a (paste_remainder_here)
Strongly disagree
Disagree
Neutral
Not Applicable?
If neutral or not applicable, please explain:
After the meeting, I felt I had a (paste_remainder_here)
Strongly disagree
Disagree
Neutral
Not Applicable?
If neutral or not applicable, please explain:
...then click 'Save Selection as Template'
New Field
New Group Field
Tabs
Page Check Code
Cut
Copy
Paste
Delete
Apply Default Fonts
Align Selection in One (1) Column Ctrl+1
Align Selection in Two (2) Columns Ctrl+2
Align Selection in Three (3) Columns Ctrl+3
Align Selection in Four (4) Columns Ctrl+4
Align Selection in Five (5) Columns Ctrl+5
Align Selection on Row Ctrl+Right
Make Same Width (Use Maximum) Ctrl+W
Make Same Width (Use Minimum) Ctrl+Shift+W
Make Same Height (Use Maximum) Ctrl+H
Make Same Height (Use Minimum) Ctrl+Shift+H
Make Same Size (Use Maximum) Ctrl+S
Make Same Size (Use Minimum) Ctrl+Shift+S
Save Selection as Template

Figure 7.115: Create New Field Template

Note: Right clicking on a field or field group will not display the proper menu options to create a field template. Right click on a blank portion of the canvas only.

4. Enter a **Template Name** in the **Add Template** dialog box.


Please select the most appropriate response for each statement which corresponds most closely to your desired response.

Using the [specific_question_here]

Template Name: Likert Questions - 2 sets

Figure 7.116: Field Template Name

5. Click **OK**. The field template appears under **Templates > Fields** in the Project Explorer.

Using Field Templates

To use a Field Template to create a new set of fields, follow these steps:

1. Locate the **Field Template** in the Project Explorer.

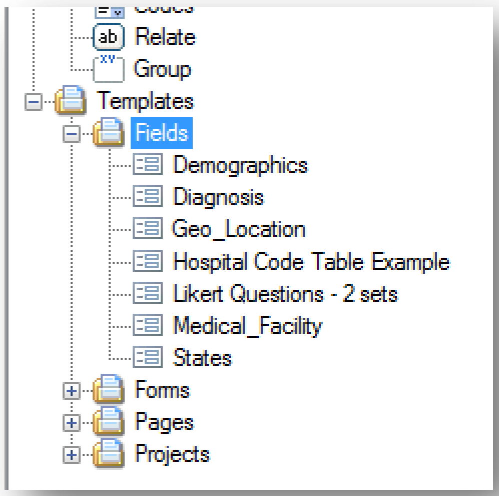
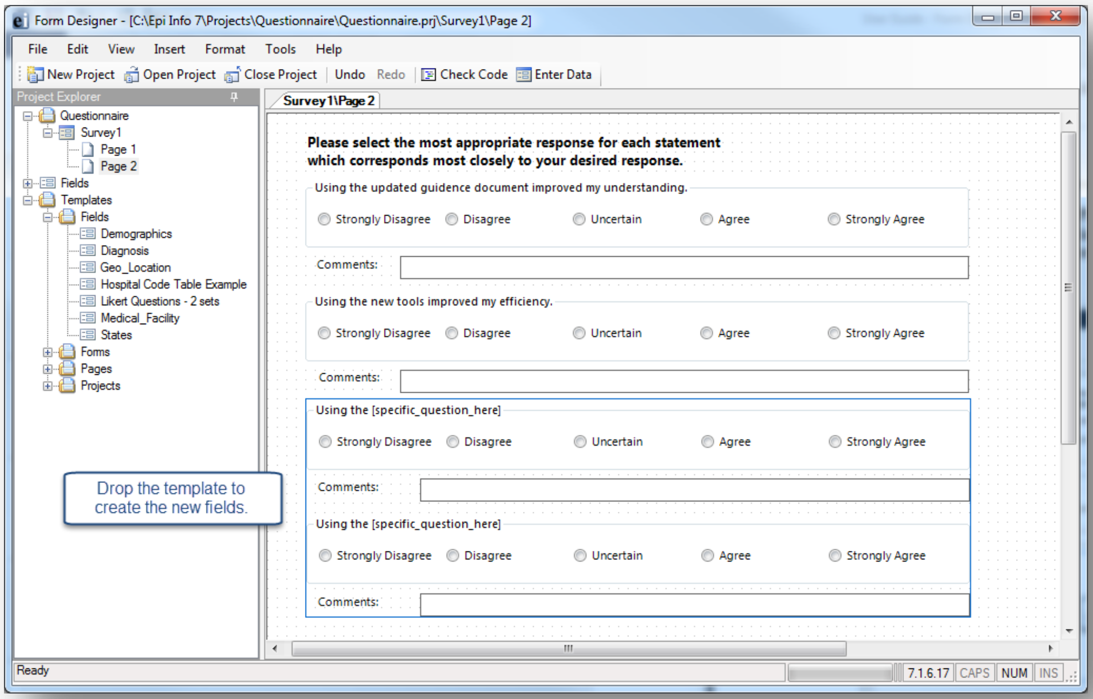


Figure 7.117: Field Template drop-down list

2. If you cannot find the template you need, but you know it is saved somewhere, use the **File > Get Template…** feature to browse for the template.

3. **Click** and **drag** the desired template from the **Project Explorer** onto the page in the appropriate location. As you drag the template over the canvas, you will see a blue rectangle showing the approximate footprint that the fields will occupy when created.

4. **Drop** the template where new field(s) should be placed. When the template is released, Epi Info™ 7 will create the new fields and copy the associated Check Code into the Check Code Editor.



Please select the most appropriate response for each statement which corresponds most closely to your desired response.

Using the updated guidence document improved my understanding.
Strongly Disagree Disagree Uncertain Agree Strongly Agree
Comments:

Using the new tools improved my efficiency.
Strongly Disagree Disagree Uncertain Agree Strongly Agree
Comments:

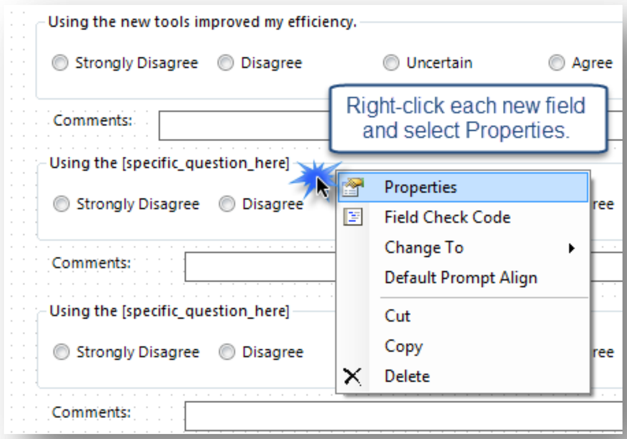
Using the [specific_question_here]
Strongly Disagree Disagree Uncertain Agree Strongly Agree
Comments:

Using the [specific_question_here]
Strongly Disagree Disagree Uncertain Agree Strongly Agree
Comments:

Figure 7.118: Drag Field Template to Canvas

Epi Info™ 7 creates fields from Field Templates using the field name specified in the template UNLESS a field by that name already exists on the form, perhaps on another page. To avoid duplicate field names within a given form, Epi Info™ 7 automatically appends a number to the field name of any field that already exists in the same Epi Info Form. Unfortunately, this has the side effect of potentially breaking the Check Code that may exist for the new fields. The field names within the Check Code are not synchronized. Therefore, the Check Code must be manually updated with any new field names.

5. For each new field created from the template, right click the field and select Properties. The **Field Property** dialog box for the selected field opens. Notice the Field Name.



Using the new tools improved my efficiency.

Strongly Disagree Disagree Uncertain Agree

Comments:

Right-click each new field and select Properties.

Using the [specific_question_here]

Strongly Disagree Disagree

Comments:

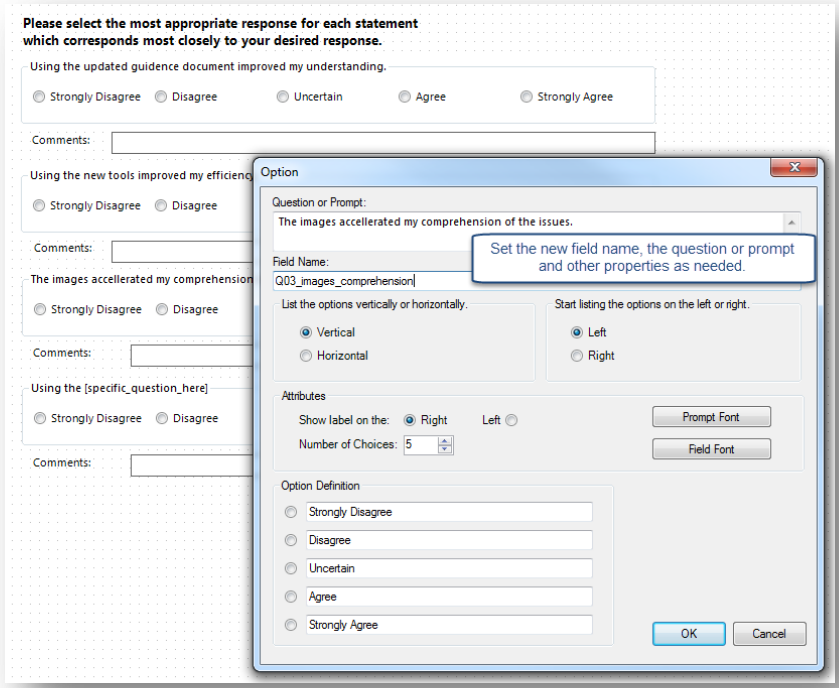
Using the [specific_question_here]

Strongly Disagree Disagree

Comments:

Figure 7.119: Open the Properties for new fields.

6. If a data table has not been created for the form, edit the **Field Name** and other properties as needed.



Please select the most appropriate response for each statement which corresponds most closely to your desired response.

Using the updated guidence document improved my understanding.

Strongly Disagree Disagree Uncertain Agree Strongly Agree

Comments:

Using the new tools improved my efficienc

The images accellerated my comprehension

Using the [specific_question_here]

Option

Question or Prompt: The images accellerated my comprehension of the issues.

Field Name: Q03_images_comprehension

Set the new field name, the question or prompt and other properties as needed.

List the options vertically or horizontally. [✓ Vertical] Horizontal

Start listing the options on the left or right. [✓ Left] Right

Attributes

Show label on the: [✓ Right] Left

Number of Choices: 5

Option Definition: Strongly Disagree; Disagree; Uncertain; Agree; Strongly Agree

Figure 7.120: Customize the new fields' names, question, and properties.

Sample Field Templates for Demonstration

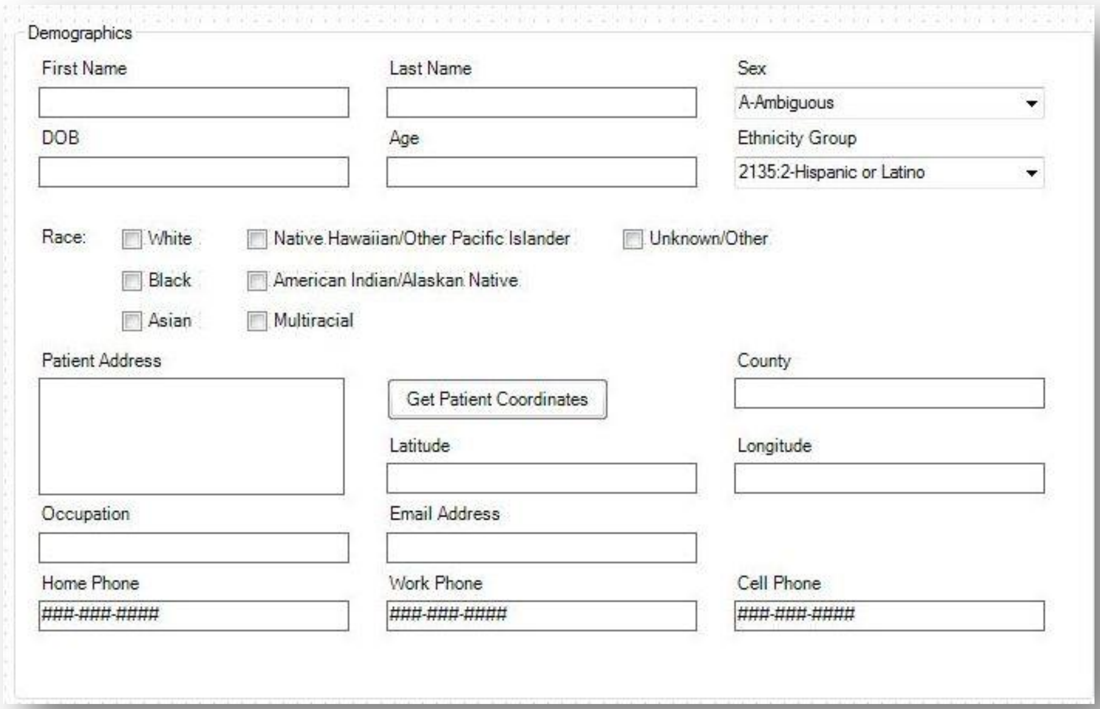
There are several field templates that are packaged with Epi Info™ 7 for demonstration purposes. While these fields could be used "As Is" for data collection, these field templates and the Check Code they contain are meant to illustrate the features and functions of Epi Info™ 7.

Demographics

The demographics template has common fields used to for patient information. Some of the fields originate from the Public Health Information Network (PHIN) vocabulary set such as Sex and Ethnicity Group.

Field Types Illustrated:

- Text
- Comment Legal Drop Down
- Phone Number
- Checkbox
- Group
- Number
- Multiline
- Command Button



Demographics

First Name | Last Name | Sex: A-Ambiguous

DOB | Age | Ethnicity Group: 2135:2-Hispanic or Latino

Race: White, Black, Asian, Native Hawaiian/Other Pacific Islander, American Indian/Alaskan Native, Multiracial, Unknown/Other

Patient Address | Get Patient Coordinates | County

Latitude | Longitude

Occupation | Email Address

Home Phone: ###-###-#### | Work Phone: ###-###-#### | Cell Phone: ###-###-####

Figure 7.121: Demographics Field Template

Diagnosis

The diagnosis template contains multiple Yes-No fields related to common patient diagnoses. The template also includes text fields for collecting additional information.

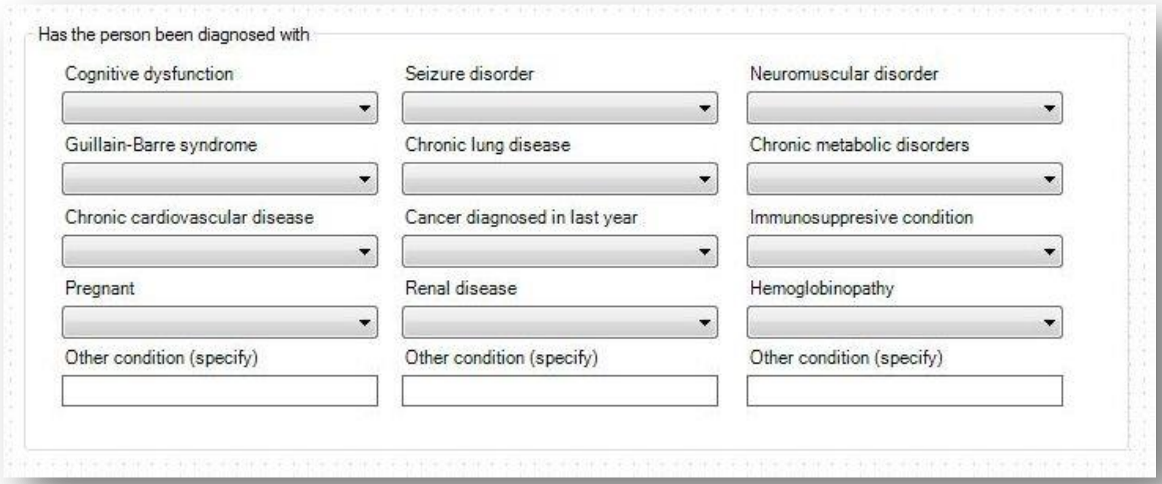

Has the person been diagnosed with
Cognitive dysfunction
Seizure disorder
Neuromuscular disorder
Guillain-Barre syndrome
Chronic lung disease
Chronic metabolic disorders
Chronic cardiovascular disease
Cancer diagnosed in last year
Immunosuppresive condition
Pregnant
Renal disease
Hemoglobinopathy
Other condition (specify)
Other condition (specify)
Other condition (specify)

Figure 7.122: Diagnosis Field Template

Geo-location

The geo-location field template contains fields for Address, Latitude, and Longitude. It also has a command button, **Get Coordinates** for which Check Code was written using the GEOCODE command. This command uses a geolocation service to populate the **Latitude** and **Longitude** fields with coordinates based on the information entered into the **Address** field. These coordinates can then be used in mapping functions in the Epi Info™ Maps tool.

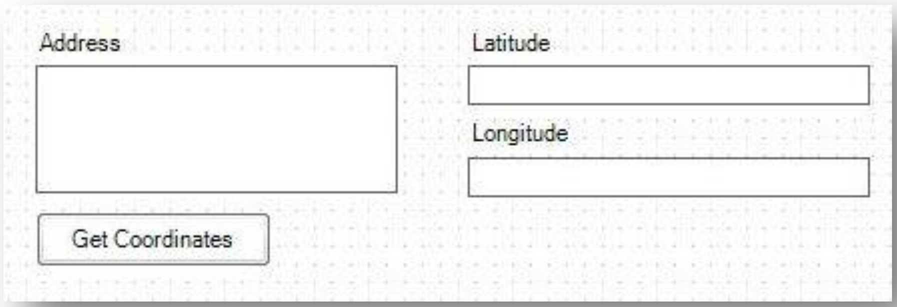
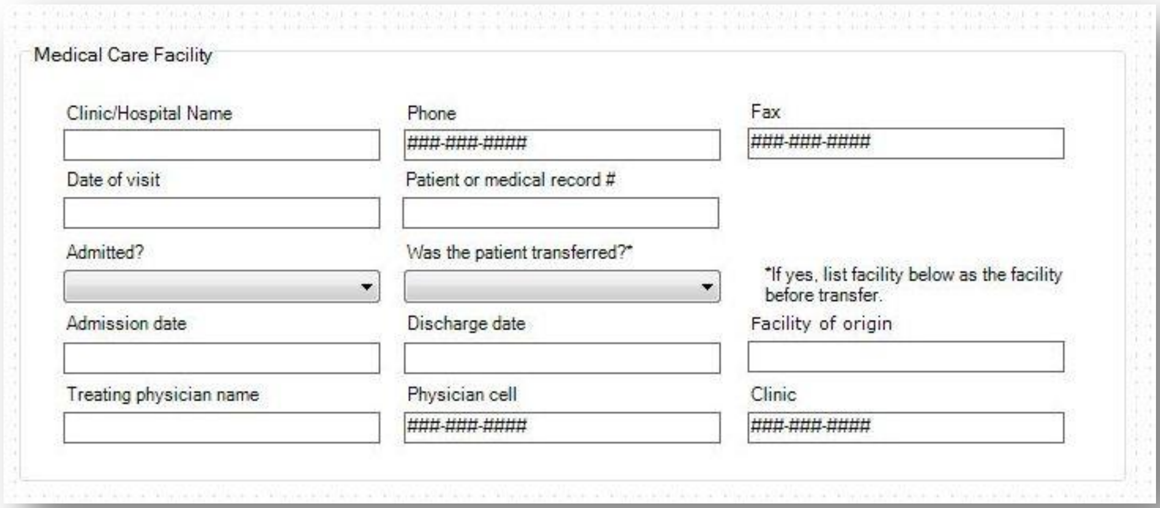


Figure 7.123: Geo_Location Field Template

Medical Facility

The Medical Facility template contains fields commonly used to collect medical facility information.

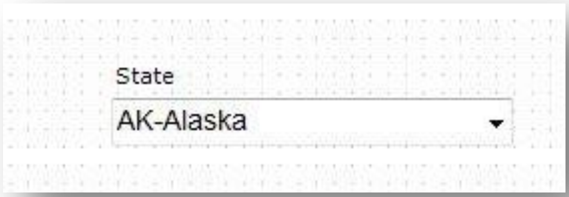

Medical Care Facility

Clinic/Hospital Name	Phone ###-###-####	Fax ###-###-####
Date of visit	Patient or medical record #	
Admitted?	Was the patient transferred?*	*If yes, list facility below as the facility before transfer.
Admission date	Discharge date	Facility of origin
Treating physician name	Physician cell ###-###-####	Clinic ###-###-####

Figure 7.124: Medical Facility Field Template

States

The States field template is an example of a comment legal field. Comment legal fields store only the code or abbreviation in the database, but the full description is shown in the drop-down list. The data source for this field includes all state and territory names and abbreviations for the United States of America in sorted order (alphabetical).


State

AK-Alaska

Figure 7.125: States Field Template

Additional Functionality in Form Designer

Insert a Line

It is sometimes helpful to add a horizontal line between groups of fields to give a visual separation between sections of the page. A line can be created using the Label/Title field.

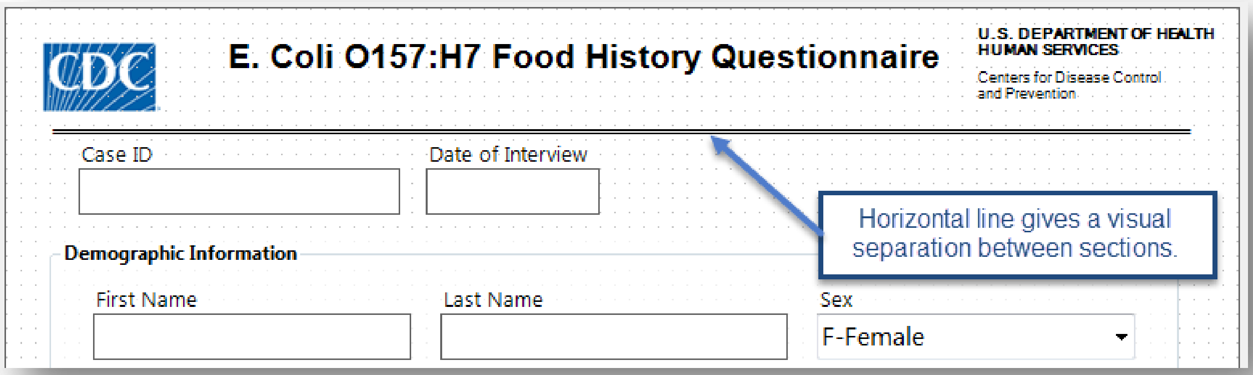

CDC
E. Coli O157:H7 Food History Questionnaire
U.S. DEPARTMENT OF HEALTH HUMAN SERVICES
Centers for Disease Control and Prevention
Case ID
Date of Interview
Demographic Information
First Name
Last Name
Sex
F-Female
Horizontal line gives a visual separation between sections.

Figure 7.126: Horizontal Line Functionality

To create a line on your page, follow these steps:

1. Right click on the canvas where you want to add a line. Select **New Field > Label/Title**.
2. In the Question or Prompt field, hold the **SHIFT key** and type an **underscore** to create a line.
3. Click the **Font** button. The Font dialog box opens.
4. Select a font **size** and **bold**.
5. Click **OK**.
6. Create a **Field Name** for the label field.
7. Click **OK**. The line appears in the form.
8. The line can be resized, moved, copied and pasted as needed.

Upgrade Project

To use a project from a previous version of Epi Info™ in Epi Info™ 7, the project must be upgraded to the proper format. The upgrade project tool allows users to browse and identify previous projects and upgrade the projects for use in Epi Info™ 7. Select **Tools > Upgrade Project > Epi Info 3.5x (.MDB).**

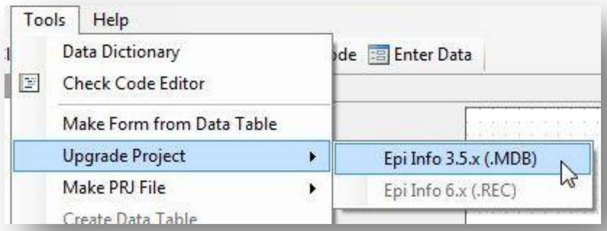


Figure 7.127: Upgrade Project

Browse and select the desired project. Click **Open**. The project is upgraded to Epi Info™ 7 and is available for use.

Make a Project File (PRJ)

Occasionally, you may receive an Epi Info™ 7 database--either the Access .MDB file or the location of the database on a SQL Server. If this database is by itself, and if you do not have the corresponding Project File (.prj), then Epi Info™ 7 will not be able to work with it until you create a new Project File.

To create a new Project File for a given Epi Info™ 7 database, follow these steps:

1. From the Form Designer Tools menu, select **Make PRJ File.**

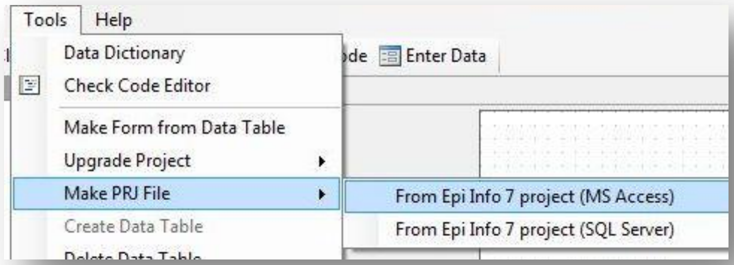


Figure 7.128: Make PRJ file

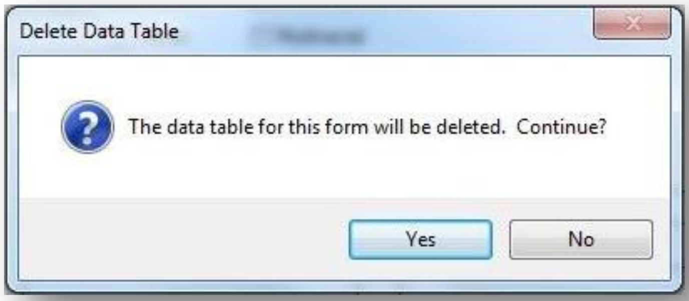
2. If the database you are attempting to use is a Microsoft Access database, .MDB, then select **From Epi Info™ 7 project (MS Access).**
   For a SQL Server database, select **From Epi Info™ 7 project (SQL Server)**.
3. For Access files, browse for the desired MDB file and click **Open**. A new project file is created from the source MDB. The original MDB is not changed so it can be opened in the older version after the migraion, however, any changes made to that original database will not be in the new upgraded version.

4. For SQL Server databases, you will be asked for the connection information for the SQL Server. See your SQL Server Database Administrator for the necessary connection information.

Delete an Existing Data Table Without Deleting the Form

To delete an existing data table without deleting the form, perform the following steps:

1. From the Form Designer toolbar, select **Tools > Delete Data Table**. The Form Designer warning message appears.


Delete Data Table

The data table for this form will be deleted. Continue?

Figure 7.129: Delete Data Table

2. Click **Yes**. The data table associated with your form is deleted. The form remains intact.

   - If the data table is deleted, any entered data associated with the form are deleted from the project. Be absolutely sure you do not need the records.

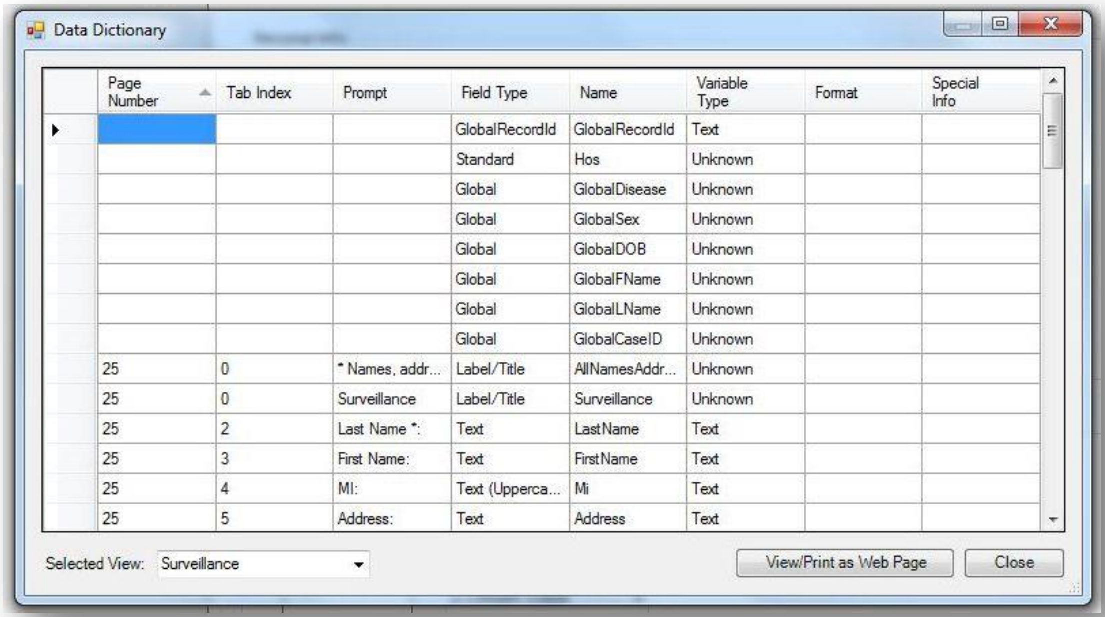
Note: This function should be used only if the data is expendable. This action is permanent and irreversible.

View a Data Dictionary

From the Form Designer toolbar, select. The resulting grid shows the name of the fields in the “Name” column. These names correspond to the column names in the data tables that you will need to use when performing analyses.

The Data Dictionary displays form(s) and defined variables for an open project. Fields or variables are sorted and displayed by page number in the form with defined variables appearing at the end of the listing. Information retrieved from the form includes Page Number, Tab Index, Prompt, Field Type, Name, Variable Type, Format, and Special Info.

From the Form Designer toolbar, select **Tools > Data Dictionary**. The Data Dictionary table appears on the canvas.



Page Number	Tab Index	Prompt	Field Type	Name	Variable Type	Format	Special Info
			GlobalRecordId	GlobalRecordId	Text		
			Standard	Hos	Unknown		
			Global	GlobalDisease	Unknown		
			Global	GlobalSex	Unknown		
			Global	GlobalDOB	Unknown		
			Global	GlobalFName	Unknown		
			Global	GlobalLName	Unknown		
			Global	GlobalCaseID	Unknown		
25	0	* Names, addr...	Label/Title	AllNamesAddr...	Unknown		
25	0	Surveillance	Label/Title	Surveillance	Unknown		
25	2	Last Name *:	Text	LastName	Text		
25	3	First Name:	Text	FirstName	Text		
25	4	MI:	Text (Upperca...	Mi	Text		
25	5	Address:	Text	Address	Text		

Selected View: Surveillance

Figure 7.130: Data Dictionary

- Page Number values are developed each time a page is added and corresponds to the page location in the Project Explorer.
- Tab Index corresponds to the tab order for each field.
- Prompt, Field Type, Name and Variable type correspond to the specifications in the Field Definition dialog box for each variable.
- Format column values include selected patterns for number, date, date-time, phone number (anything with a pattern or format) fields and sorted for combo boxes. Combo box is a combination of list box and a drop-down list (legal values, combo).
- Special Info column values include all properties available from the Field Definition dialog box. Properties include Read Only, Legal Value, Repeat Last, Code Table, Groups, Required, Range, and Image Size. The Special Info column also holds the defined variables values of Standard, Global, or Permanent. The Special Info column includes information on related fields to indicate whether they contain one record or an unlimited number of records. This is developed when the related field is created. The default is Unlimited Records. If the Return to the Parent Form after One Record has been Entered box is checked, the format will appear as one record.

To view a data table located in another form:

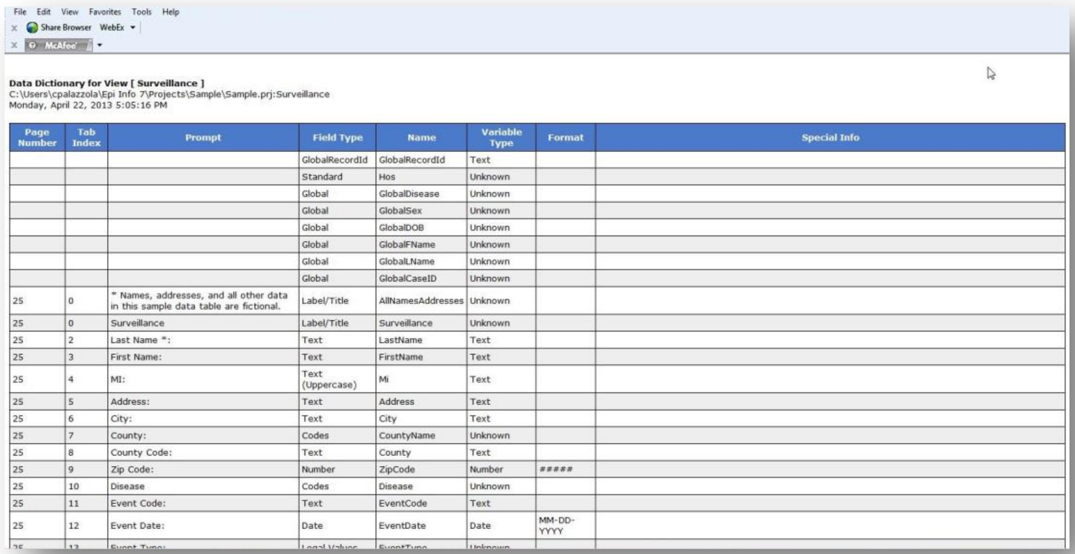
1. Click **Selected View**. The Selected View drop-down list opens.
2. Select the **form** to view.
3. The Data Dictionary for the selected form opens. Note that only the Data Dictionary for the selected form opens.

To open the Data Dictionary as an HTML page inside the browser window:

1. Click **View/Print as Web Page**.

Note: From the browser, the data can be printed with **File > Print,** or saved with **File > Save As**.

2. Right click on the **HTML page** to show the context menu. You can export the data directly to an Excel spreadsheet or print the document.



Data Dictionary for View [Surveillance]
C:\Users\cpalazzola\Epi Info 7\Projects\Sample\Sample.prj:Surveillance
Monday, April 22, 2013 5:05:16 PM

Page Number	Tab Index	Prompt	Field Type	Name	Variable Type	Format	Special Info
			GlobalRecordId	GlobalRecordId	Text		
			Standard	Hos	Unknown		
			Global	GlobalDisease	Unknown		
			Global	GlobalSex	Unknown		
			Global	GlobalDOB	Unknown		
			Global	GlobalFName	Unknown		
			Global	GlobalLName	Unknown		
			Global	GlobalCaseID	Unknown		
25	0	* Names, addresses, and all other data in this sample data table are fictional.	Label/Title	AllNamesAddresses	Unknown		
25	0	Surveillance	Label/Title	Surveillance	Unknown		
25	2	Last Name *:	Text	LastName	Text		
25	3	First Name:	Text	FirstName	Text		
25	4	MI:	Text (Uppercase)	Mi	Text		
25	5	Address:	Text	Address	Text		
25	6	City:	Text	City	Text		
25	7	County:	Codes	CountyName	Unknown		
25	8	County Code:	Text	County	Text		
25	9	Zip Code:	Number	ZipCode	Number	#####	
25	10	Disease	Codes	Disease	Unknown		
25	11	Event Code:	Text	EventCode	Text		
25	12	Event Date:	Date	EventDate	Date	MM-DD-YYYY	
25	13	Event Type:	Legal Values	EventType	Unknown		

Figure 7.131: Data Dictionary as an HTML Page

3. Click **Close** to exit the Data Dictionary.

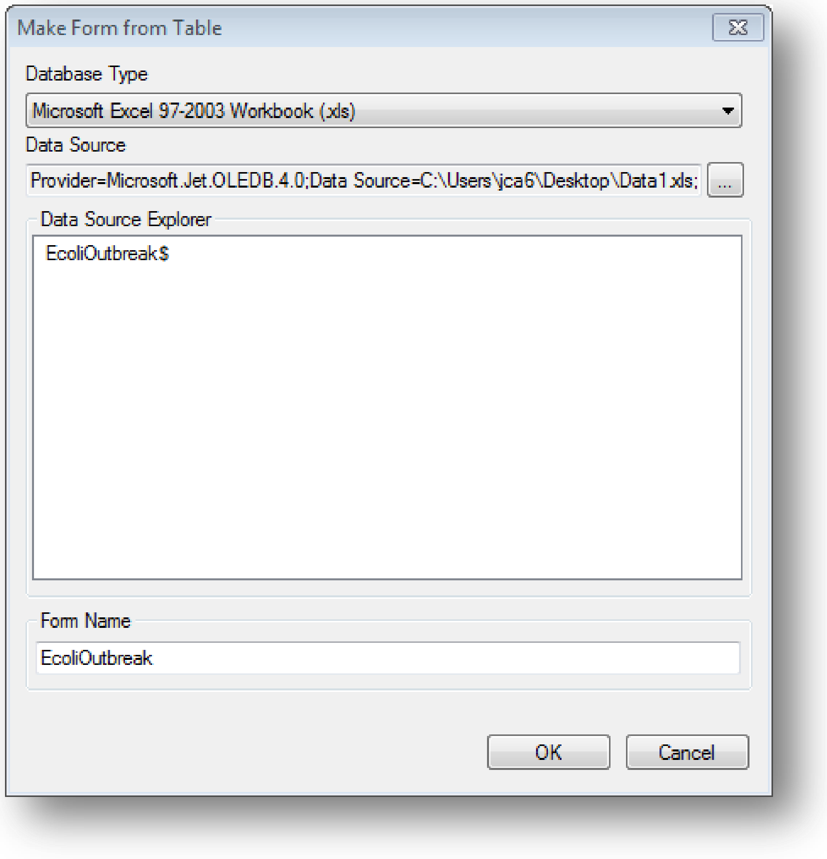
Make Form from Data Table

The Make Form from Data Table feature allows you to automatically generate an Epi Info 7 form based on an existing database file. Once completed, you will have the initial structure of a form that could be modified further if needed. Field types, code reference tables and field prompts can be specified during the process. Formats supported include MS Excel, MS Access, Flat ASCII, SQL Server and MySQL among other types.

For example, let's say that you have an Excel spreadsheet with data that you have historically maintained using MS Excel but would like to create an Epi Info project to collect the data and enjoy the benefits that can be obtained from entering data using Epi Info 7. This feature allows users to do so.

The first step will require you to create an Epi Info 7 project. When prompted for a Form Name, assign any name to the form. This initial form will be deleted eventually after the import process from the MS Excel file is completed.

1. From the Epi Info main menu, select **Utilities > Make Form from Data Table**. The **Make Form from Table** dialog opens.
2. Select the desired Database Type. In this example, we will select Microsoft MS Excel 97-2003 Workbook.
3. Navigate to the location of the file.
4. Click on the name of the spreadsheet that contains the dataset you would like to import into Epi Info 7.
5. Specify a form name.



Database Type
Microsoft Excel 97-2003 Workbook (.xls)
Data Source
Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\Users\jca6\Desktop\Data1.xls;
Data Source Explorer
EcoliOutbreak$
Form Name
EcoliOutbreak

Figure 7.132: Make Form from Table dialog

6. Click **OK**. The Table-to-Form dialog opens.

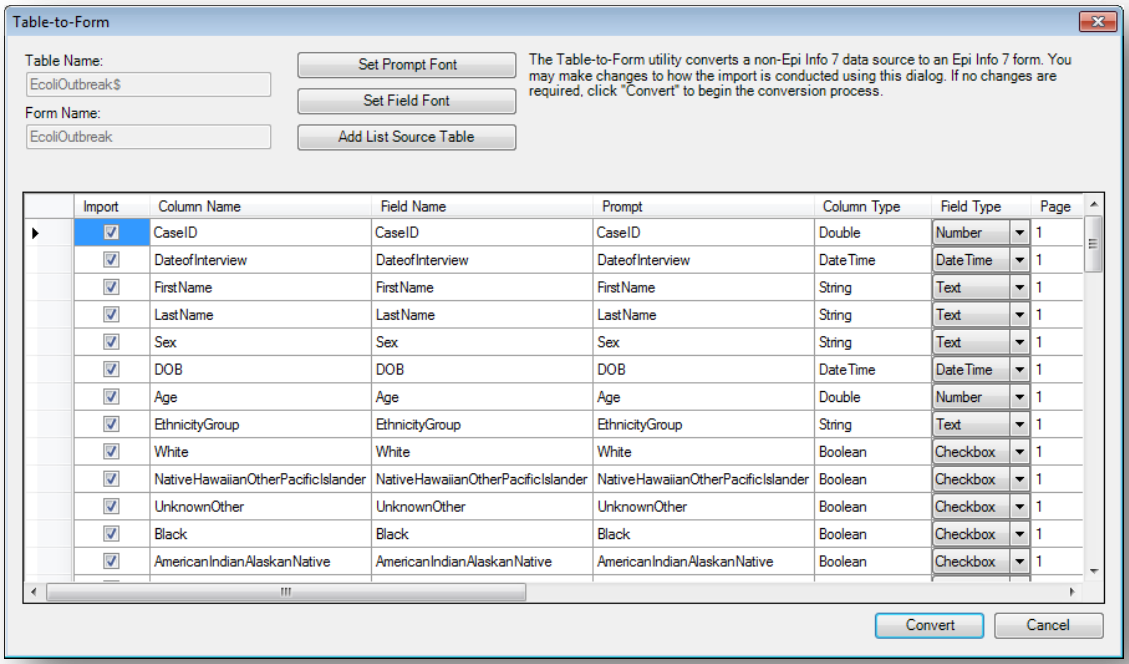


Table Name: EcoliOutbreak$

Form Name: EcoliOutbreak

The Table-to-Form utility converts a non-Epi Info 7 data source to an Epi Info 7 form. You may make changes to how the import is conducted using this dialog. If no changes are required, click "Convert" to begin the conversion process.

Import	Column Name	Field Name	Prompt	Column Type	Field Type	Page
✓	CaseID	CaseID	CaseID	Double	Number	1
✓	DateofInterview	DateofInterview	DateofInterview	DateTime	DateTime	1
✓	FirstName	FirstName	FirstName	String	Text	1
✓	LastName	LastName	LastName	String	Text	1
✓	Sex	Sex	Sex	String	Text	1
✓	DOB	DOB	DOB	DateTime	DateTime	1
✓	Age	Age	Age	Double	Number	1
✓	EthnicityGroup	EthnicityGroup	EthnicityGroup	String	Text	1
✓	White	White	White	Boolean	Checkbox	1
✓	NativeHawaiianOtherPacificIslander	NativeHawaiianOtherPacificIslander	NativeHawaiianOtherPacificIslander	Boolean	Checkbox	1
✓	UnknownOther	UnknownOther	UnknownOther	Boolean	Checkbox	1
✓	Black	Black	Black	Boolean	Checkbox	1
✓	AmericanIndianAlaskanNative	AmericanIndianAlaskanNative	AmericanIndianAlaskanNative	Boolean	Checkbox	1

Figure 7.133: Table-to-Form dialog

7. Click **Set Prompt Font** and **Set Field Font** to set the fonts for the question/prompts and the input fields, respectively.
8. If any of the columns in the source table are to become a Legal Value field type, then click **Add List Source Table** and browse for the table that contains the list of valid values. This will be needed below for the **ListSourceTableName** column.

A series of columns are displayed. Here is a description of each column:

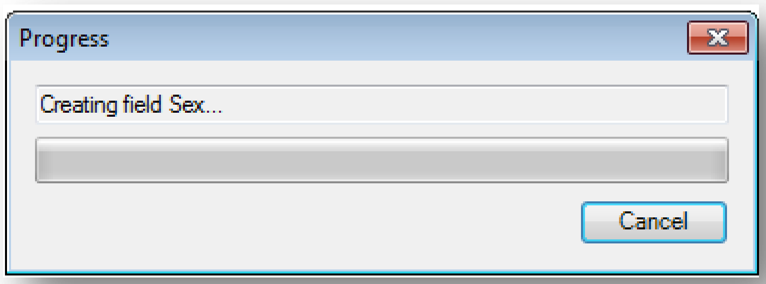
- **Import-** This check box allows you to select which fields should be imported into the form and turn off the import for other fields.. By default, all fields are checked.
- **Column Name**- This specifies the column name from the source table. The column name is initially used for the field name and prompt.
- **Field Name**- The field name is initially the same as the column name, but you can change it now. If the field name is an Epi Info reserved word, the Table-to-Form process will change it automatically by adding a suffix of "_RW". You may want to change this name to something more relavent. ***Note: Be sure these field names are adequate before starting the conversion because they cannot be changed after the data tables are created.***
- **Prompt**- The prompt is initially the same as the field name. You can change it now here, or later by visiting the properties for the field.
- **Column Type** – This is the type of data that Epi Info™ 7 detected.
- **Field Type** – This specifies the type of field that probably corresponds to the Column Type. For example, if Epi Info™ 7 detects string data for the column type,

then it will suggest the type of field to hold this data might be a Text field. You can override this decision if you know what the data repesent. For example, if the column contains only "Yes" and "No" values, Epi Info™ may suggest the Text field, but you may decide the YesNo field type is more appropriate. Available field types include: Text, Multiline, Number, YesNo, Checkbox, Date, Date-Time, Time, and Legal Values.

- **Page**- For tables having many columns, Epi Info™ will place the fields on the form based on the order they appear on the source table. It will stack as many fields as will fit on the first page, then it will continue to add pages and stacking fields as needed to accommodate the all of the columns. You can use the Page column to instruct Epi Info™ to keep certain fields together on a specific page.
  Note: There is a limit of about 250 fields on any given page.
- **Tab**- This specifies the order of the field within the page during data entry.
- **Tab stop**- If data entry will occur for a field, then this should remain checked. Some fields are automatically calculated, so these would not need a Tab stop and therefore the Tab stop checkbox should be unchecked. When Tab Stop is unchecked, the cursor will not move into the field during data entry.
- **Read only**- This check box sets the Read Only attribute of the field. If checked, the field will be disabled. By default, it is unchecked.
- **Required** – Fields where data entry is manditory should have this Required attribute checked. By default, it is unchecked.
- **Repeat last** - When checked, the Repeat Last attribute causes the value of the most recently viewed record to become the pre-populated values when a new record is created. By default, it is unchecked.
- **Range** - Allows for a specified value between one value and another. During data entry, if you attempt to enter a value outside the specified range, you'll get a warning message that the value is out of range. Missing values are accepted. By default, it is unchecked. If you check the Range check box, then also specify the Lower and Upper ranges as described below.
- **Lower**- Specify the smallest or lowest value that can be entered into the field.
- **Upper** - Specify the largest or highest value that can be accepted by the field.
- **List Source Table Name** – If a reference table has been imported using the *Add List Source Table* option, the field can be linked to one of the imported tables. This only applies to Legal Value fields.
- **List Source Text Column Name** – If more than one column is available on the imported reference table, specify which column name should be used to map to the field from within the table.
- **List Source Table**- Fields can be removed from the View Fields list using the **back single arrow** or **back double arrows**.

9. Click **Convert** to start the import process.

The process of creating the form and importing the process begins. Once the process is completed, the message "Table to form import process complete" message appears.



Creating field Sex...

Figure 7.134: In Progress dialog window

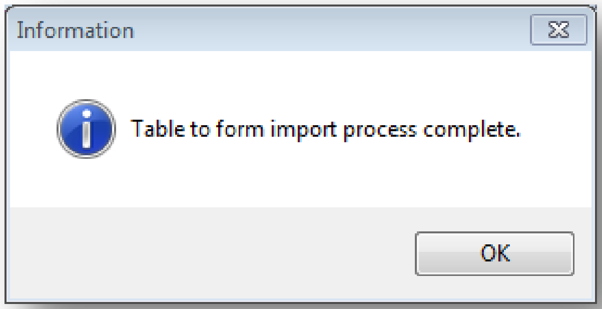
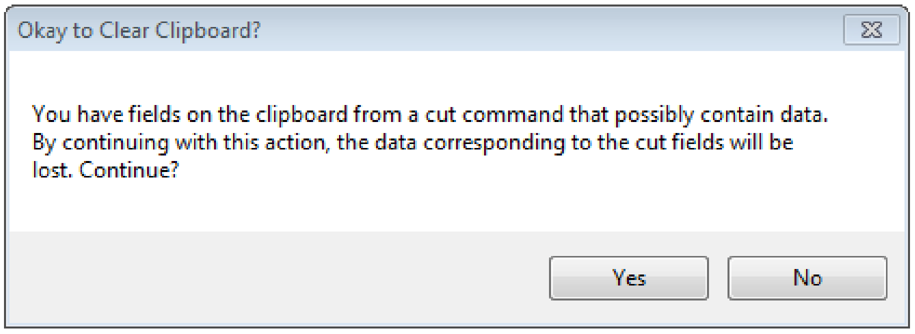


Table to form import process complete.

Figure 7.135: Process completed dialog window

After the import process has completed, you can make some changes to the form such as alignment, font, and adding labels and titles. However, do not cut and paste fields from one page to another. If you intend to move fields across pages after the import, we strongly recommended that you first **Delete Data Tables Without Deleting the Form**, move the fields to the necessary pages, then **Use the MERGE Command** in Classic Analysis to merge the data into the revised form.



You have fields on the clipboard from a cut command that possibly contain data. By continuing with this action, the data corresponding to the cut fields will be lost. Continue?

Figure 7.136: Cutting and Pasting fields

3. Check Code: Customizing the Data Entry Process

Introduction

The Check Code tool allows users to customize the data entry process. It is useful to check for errors during the data entry process, to do automatic calculations in fields, and to skip over parts of the questionnaire if certain conditions are met. Check Code makes it possible to instruct the Enter tool to perform such operations automatically. By using Check Code, you can protect data against many common types of errors by setting rules for data entry. Check Code is created using the Check Code Editor.

Examples of operations that can be performed in Check Code include:

- Displaying messages that appear to be part of the questionnaire
- Calculating fields from mathematical operations
- Checking one or more fields for relationships (e.g., making sure birth date is earlier than current date)
- Checking fields for inconsistencies (e.g., male pregnancies)
- Displaying error messages due to improper entries in a field
- Complex statistics or other operations that are written in other languages
- Automatic indexing of fields for faster searching
- Automatic searches during data entry

An operation can be performed automatically each time data is entered in a field or conditionally when a certain value is entered. Check code is optional and is designed to allow the user to provide customized data entry processes.

Accessing Check Code Program Editor

To navigate to the **Check Code Program Editor**, click the **Check Code button** in the Form Designer tool bar after opening your Epi Info 7 project.

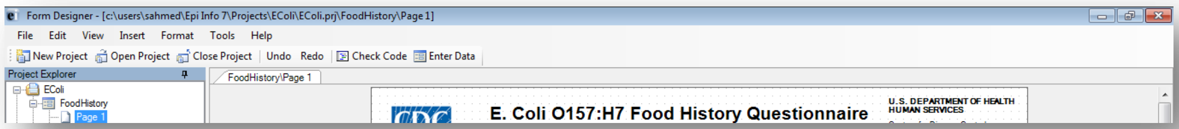

Form Designer - [c:\users\sahmed\Epi Info 7\Projects\EColi\EColi.prj\FoodHistory\Page 1]
File
Edit
View
Insert
Format
Tools
Help
New Project
Open Project
Close Project
Undo
Redo
Check Code
Enter Data
Project Explorer
EColi
FoodHistory
Page 1
FoodHistory\Page 1
E. Coli O157:H7 Food History Questionnaire
U.S. DEPARTMENT OF HEALTH HUMAN SERVICES

Figure 7.137: Check Code button

Alternatively, select **Tools > Check Code Editor** from the **Form Designer** navigation menu.

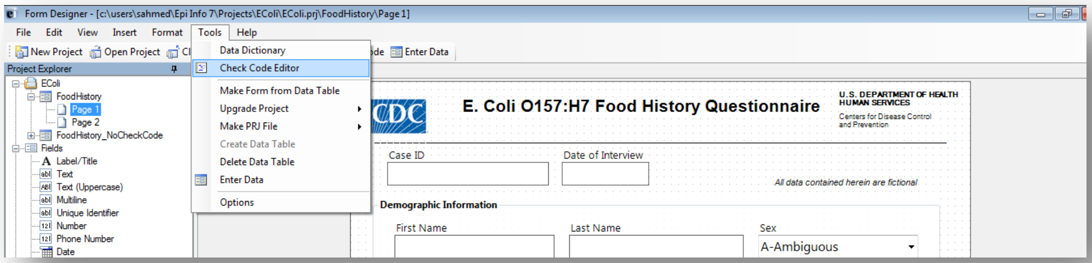
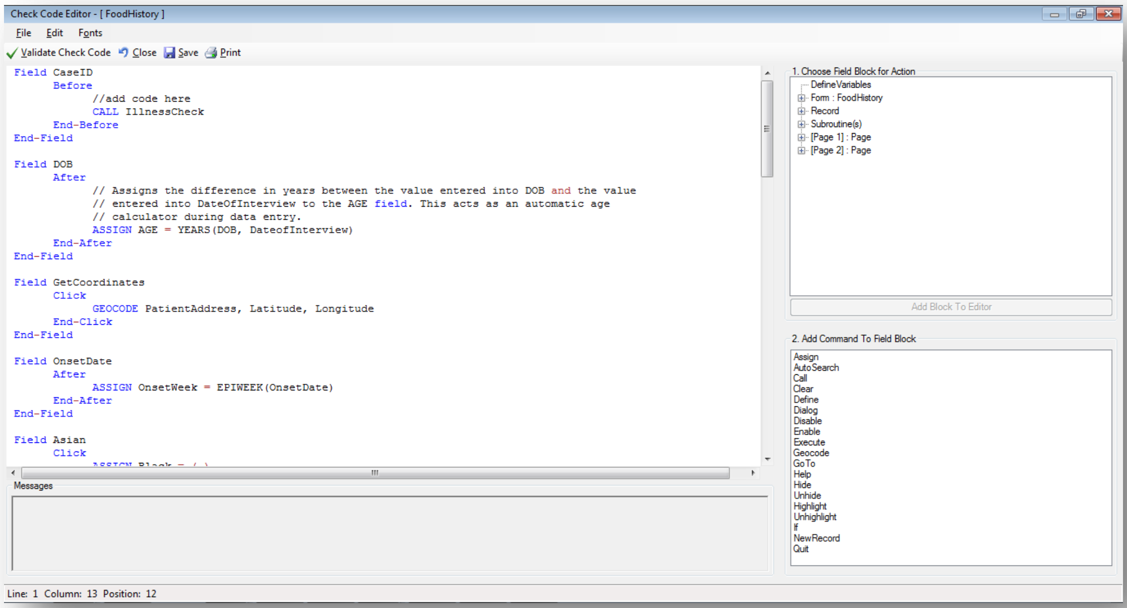


Figure 7.138: Form Designer Menu

Navigate the Check Code Program Editor

The Check Code Editor window contains four sections:

- **Choose Field Block for Action**
- **Add Command to Field Block**
- **Program Editor**
- **Messages**



```
Field CaseID
      Before
            //add code here
            CALL IllnessCheck
      End-Before
End-Field

Field DOB
      After
            // Assigns the difference in years between the value entered into DOB and the value
            // entered into DateOfInterview to the AGE field. This acts as an automatic age
            // calculator during data entry.
            ASSIGN AGE = YEARS(DOB, DateofInterview)
      End-After
End-Field

Field GetCoordinates
      Click
            GEOCODE PatientAddress, Latitude, Longitude
      End-Click
End-Field

Field OnsetDate
      After
            ASSIGN OnsetWeek = EPIWEEK(OnsetDate)
      End-After
End-Field

Field Asian
      Click
```

Figure 7.139: Check Code Program Editor

1. The **Choose Field for Block Action** tree allows you to select fields and sets when the actions designated by the Check Commands occur during data entry.

2. The **Add Command to Field Block** window displays all the available check commands used in the Form Designer program.

3. The **Program Editor** window displays the code generated by the commands created from the **Choose Field Block for Action** or **Add Command to Field Block** window. Code can also be typed and saved directly into the Program Editor.

4. The **Message** window alerts you of any check command problems

There are several options on the toolbar at the top left that allow you to save, edit, validate and change the font of your program editor. You can close the Check Code Editor and return to your form by clicking on the red X button at the top right of your screen, clicking on the Close button, or by pressing the F10 key.

Check Code Commands

Check commands are structured in blocks. Each block begins with a field, page or form name and ends with the word END. All commands must be within a field-name block. Commands in a block are usually activated either before or after an entry is made in the field. For some field types, blocks can also be activated when clicking on the control (i.e.

checkboxes and command buttons). The usual case is that the commands are performed after an entry has been made and the user has pressed the <ENTER> key or when the cursor has left the field automatically. This behavior can be altered by placing commands in blocks called BEFORE. The format of check code blocks is usually structured as follows:

Field VARIABLENAME

After

Check Code syntax inserted here

End-After

End-Field

- The **FIELD** parameter establishes to which field name the check code block corresponds.
- The **AFTER** parameter specifies when the action will occur. The AFTER event is executed as soon as the cursor leaves that field.
- The **END-AFTER** parameter specifies the closing of the commands to be executed, in this example, for the AFTER event. In other words, any check code inserted between the AFTER and END-AFTER section will be executed for that field after the cursor leaves the field.
- The **END-FIELD** parameter simply closes the block of commands incorporated for the corresponding field.

In the example below, we have incorporated an AFTER event for a field called DOB. The block of check code will execute the assignment of a value to the field AGE using the YEARS function. The YEARS function will calculate the difference between two date fields and provide the result in Years.

Field DOB

After

ASSIGN AGE=YEARS(DOB,SYSTEMDATE)

End-After

End-Field

Basic Check Code Command Rules

1. Check Commands must be placed in a block of commands corresponding to a variable/field in the database. Special sections are provided to execute commands before or after you display a form, page, or record.
2. Comments preceded by two forward slashes ("//") may be placed within blocks of commands and will be ignored during execution of check code.
3. Commands in a block are activated before or after you make an entry in the field. By default, commands are performed after an entry has been completed with <Enter>, <PgUp>, <PgDn>, or <Tab>, or another command causes the cursor to leave the field (e.g., GOTO). Commands can also be activated when clicking on a field or when selecting a value from a drop down list (versions 7.1.4 or higher only).
4. Check commands for each field are stored in the form in a record associated with a particular field.
5. Commands are inserted automatically through interaction with the dialog boxes. The syntax generated by the dialogs is then displayed in the Check Code Editor. Text can also be edited and saved in the Program Editor.
6. BEFORE and AFTER commands can be inserted into fields but also into a form, page or a record.

Creating a Check Code Block

1. From **Choose Field Block for Action**, select and expand the form, page, record, or field that will receive the commands. Expand by clicking the + sign.

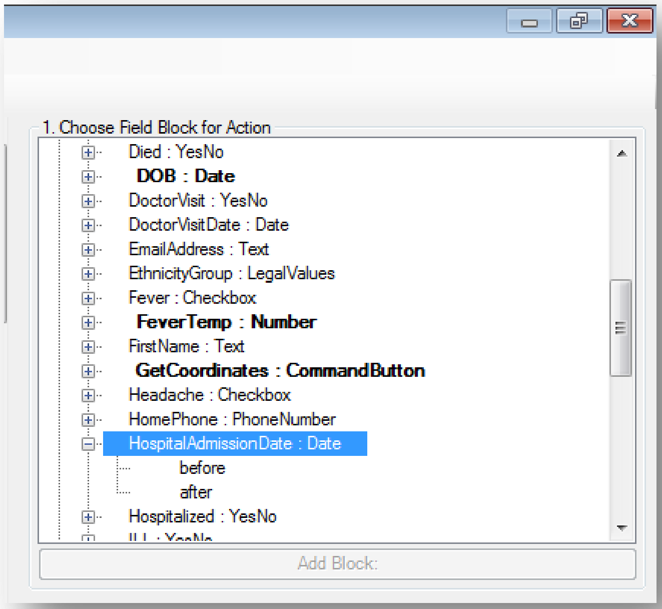


Figure 7.140: Choose Field Block for Action

2. Select whether the command will be executed **before** or **after** data entry into the form, page, record, or field by clicking in the corresponding option (in this example AFTER).

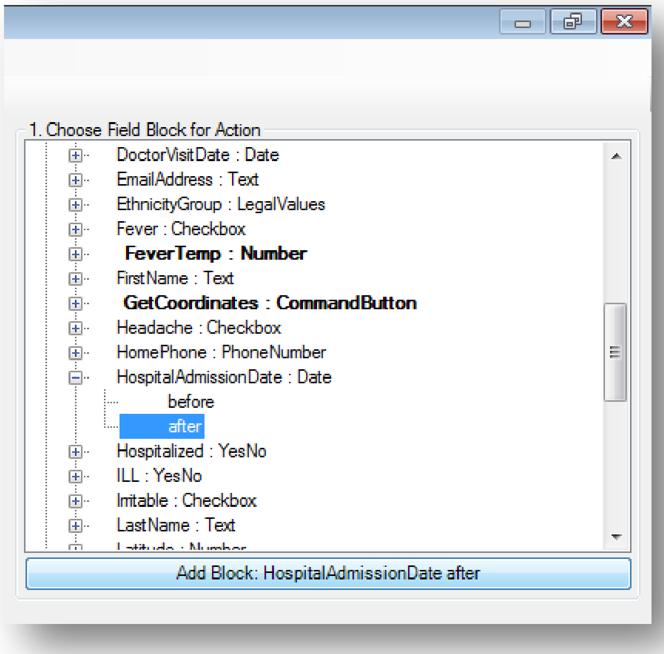
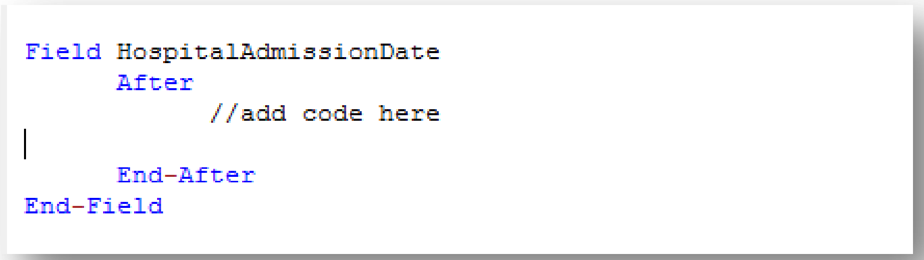


Figure 7.141: Add Block

3. You can either double click on the AFTER event or click the **Add Block** button in order to insert the block. The Check Code Block appears in the **Check Code Editor**.



```
Field HospitalAdmissionDate
      After
            //add code here

      End-After
End-Field
```

Figure 7.142: Code Block Added to Editor

After a Check Code Block has been created for a form, page, record, or field, you can insert commands within the block using the **Add Command to field Block** section.

Create a Skip Pattern with GOTO

You can create skip patterns by changing the tab order and setting a new cursor sequence in a form, or by creating Check Code using the **GOTO** command. Skip patterns can also be created based on the answers to questions using an **IF/THEN** statement. In the following example, we will add check code to move the cursor to the **Ethnicity** field after data is entered into the **DOB** field.

1. From Form Designer, Open the **Ecoli.prj project**.
2. Double click on the **FoodHistory_NoCheckCode** form.
3. Click **Check Code** or select **Tools > Check Code Editor**. The Check Code Editor opens.
4. From the **Choose Field Block for Action section,** expand the node for Page 1 to see the various fields on the first page.
5. Expand the node for the field **DOB**.

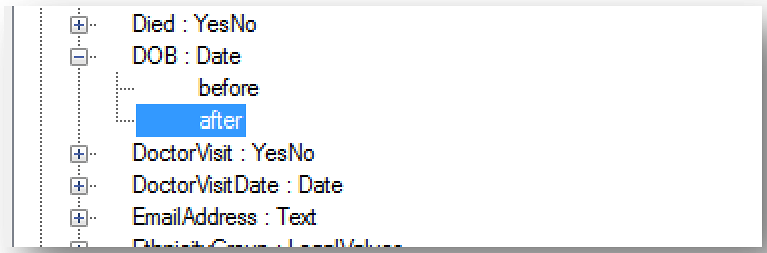
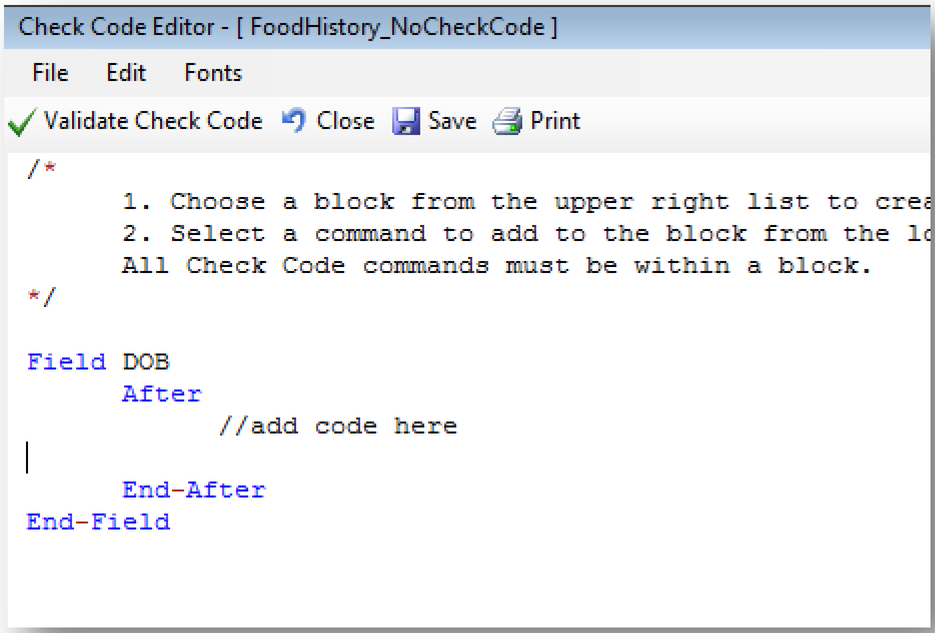


Figure 7.143: Dialog Block for Action after DOB

6. Double click on the **after** event.
7. A block of code for the **DOB** field will display in the **Check Code Editor**.



```
/*
        1. Choose a block from the upper right list to crea
        2. Select a command to add to the block from the lo
        All Check Code commands must be within a block.
*/

Field DOB
        After
                //add code here

        End-After
End-Field
```

Figure 7.144: Check Code Block for Action After DOB

8. Click **GoTo** from the **Add Command to Field Block** list box. The **GOTO** dialog box opens.
9. Select the **EthnicityGroup** field for the cursor to jump into after data has been entered in the **DOB** field. The code will run after the cursor leaves the field.

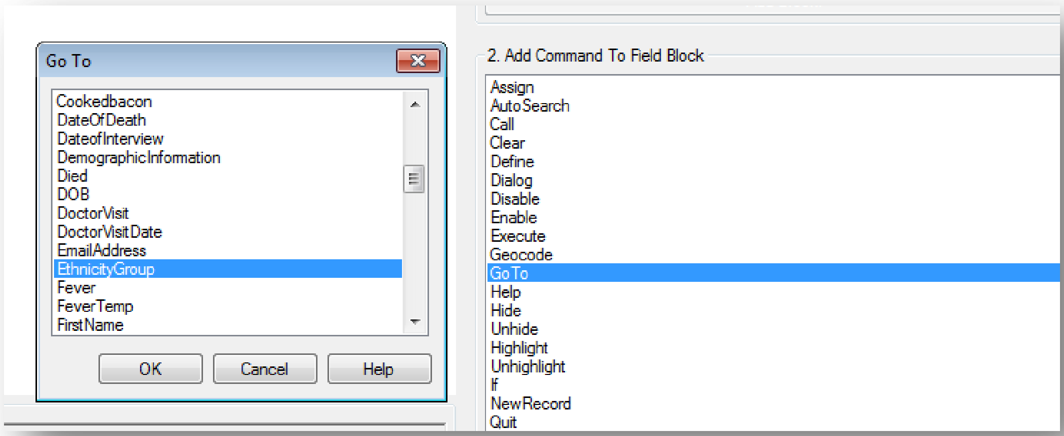
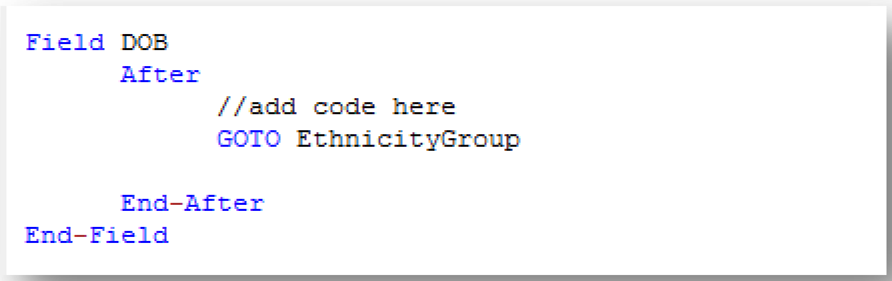


Figure 7.145: Skip Pattern **Go To** dialog box

10. Click **OK**. The code appears in the Check Code Editor.



```
Field DOB
      After
            //add code here
            GOTO EthnicityGroup

      End-After
End-Field
```

Figure 7.146: Skip Pattern Check Code Command

11. Click the **Verify Check Code** button from the Check Code Editor.
12. Click the **Save** button from the Check Code Editor.
13. Click **Close** to return to the form.

To test the skip pattern, open the form in the Enter Data tool. Use the **tab key** to ensure that upon leaving the field with the **GOTO** command, the cursor goes to the specified field.

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 form. This example creates code, which states that if the person answered No (-) for the **Hospitalized** field, then the cursor subsequently jumps to the field **Was the patient treated with antibiotics?** and skips the **Hospital Admission** date field.

1. From **Form Designer**, Open the **Ecoli.prj project**.
2. Double click on the **FoodHistory_NoCheckCode** form.
3. Click **Check Code** or select **Tools > Check Code Editor**. The Check Code Editor opens.
4. From the **Choose Field Block for Action section,** expand the node for Page 1 to see the various fields on the first page.
5. Expand the node for the **Hospitalized** field.
6. The action needs to occur after data are entered into the **Hospitalized** field.
7. Double click on the **after** event.
8. A block of code for the **Hospitalized** field is created and displayed in the Check Code Editor.

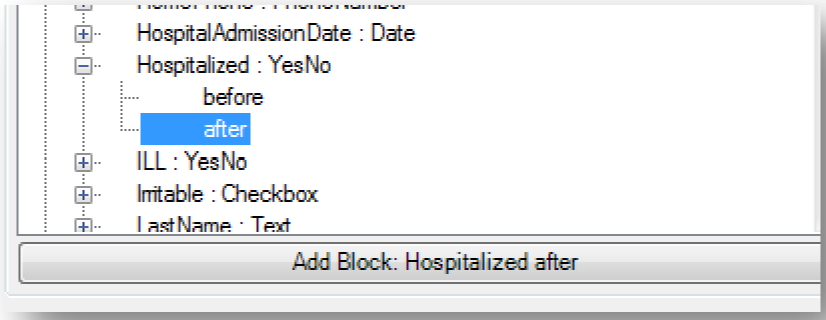
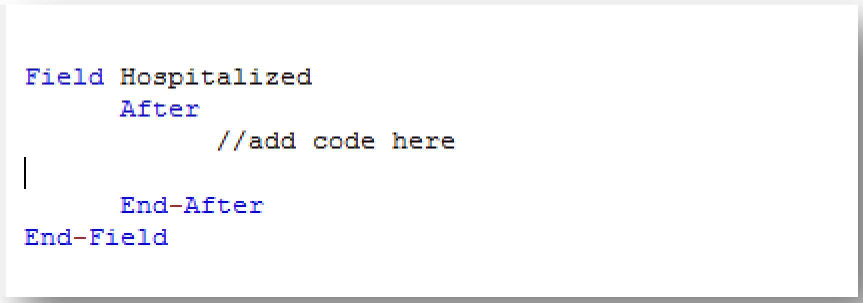


Figure 7.147: Dialog Block for Action after Hospitalized

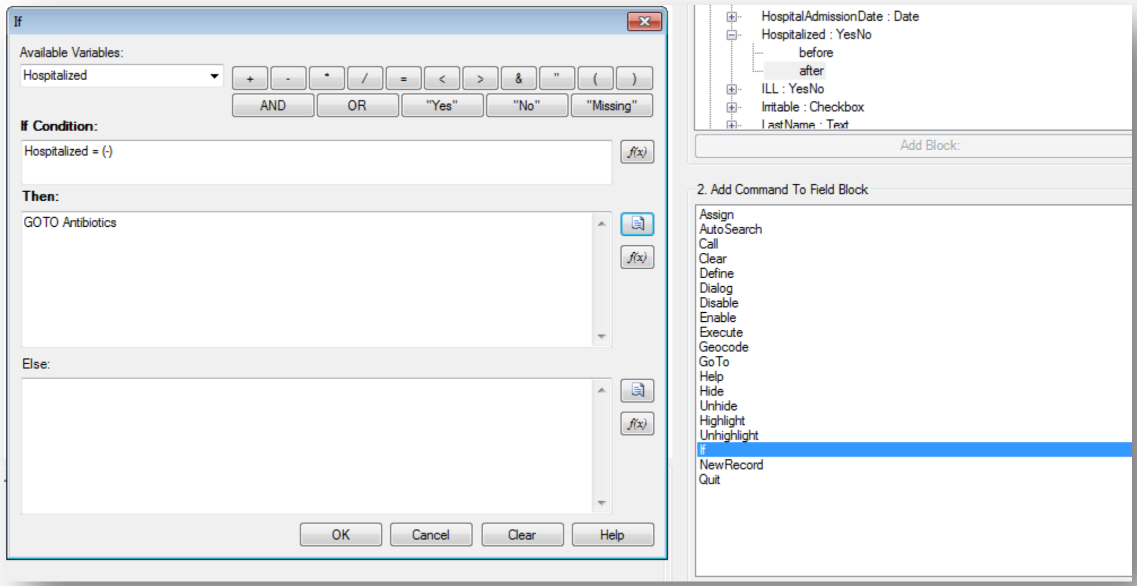
9. The code appears in the Check Code Editor.



```
Field Hospitalized
      After
            //add code here

      End-After
End-Field
```

Figure 7.148: If/Then Block Code After Hospitalized

10. Click **If** from the **Add Command to Field Block** list box. The **IF** dialog box opens.
11. From the **Available Variables** drop-down list, select the **field** to contain the action. For this example, select **Hospitalized**. The selected variable appears in the **If Condition** field.
12. From the Operators, click **=**.
13. From the Operators, click **No**. The **If Condition** field will read Hospitalized=(-).
14. Click the **Code Snippet** button in the *Then* section. A list of available commands appears.
15. From the command list, select **GoTo**. The **GOTO** dialog box opens.
16. Select the **field** for the cursor to jump to based on a *No* answer from the list of variables. For this example, select **Antibiotics**.
17. In the **GOTO** dialog box, click **OK** to return to the **IF** dialog box.



Available Variables: Hospitalized

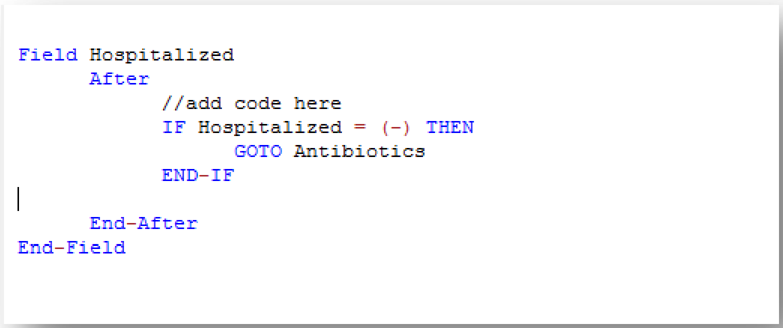
If Condition: Hospitalized = (-)

Then: GOTO Antibiotics

Else:

Figure 7.149: If/Then dialog box

18. Click **OK**. The code appears in the Check Code Editor. The example code appears as:



```
Field Hospitalized
      After
            //add code here
            IF Hospitalized = (-) THEN
                  GOTO Antibiotics
            END-IF

      End-After
End-Field
```

Figure 7.150: If/Then Check Code Command

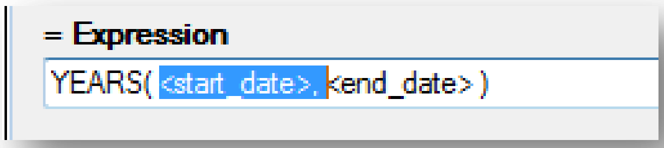
19. Click the **Verify Check Code** button from the Check Code Editor.
20. Click the **Save** button from the Check Code Editor.

Using Functions with a Date field

To program a mathematical function, use the Program Editor and the **ASSIGN** command. For example, Check Code can be created to automatically calculate the age of a respondent based on the date of birth and the date the form was completed, or the system date of the computer when data were entered.

This example uses a field called **DateOfBirth** and a field called **Age** from the **FoodHistory_NoCheckCode** form in the **Ecoli** project to demonstrate the use of the **ASSIGN** command and the function **YEARS**.

1. Click **Check Code** or select **Tools > Check Code Editor**. The Check Code Editor opens.
2. From the **Choose Field Block for Action** section, expand the node for the page where the **DateofBirth** field is located.
3. Expand the node for the **DateofBirth** field.
4. Double click on the **after** event.
5. A block of code for the **DateofBirth** field is created and displayed in the Check Code Editor.
6. Click **Assign** from the **Add Command to Field Block** list box. The **Assign** dialog box opens.
7. From the **Assign Variable** drop-down list, select the field where the calculated value should appear. For this example, select the **Age** field.
8. Click on the functions button f(x).
9. Select the **Date Functions** option.
10. Click on the **YEARS** function.
11. Double click on the <**start_date**> parameter. This will highlight that section of the command.

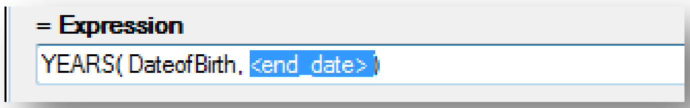


= Expression

YEARS(<start_date>, <end_date>)

Figure 7.151: Start Date Parameter

12. Select **DateOfBirth** from the list of Available Variables
13. Double click on the <**end_date**> parameter. This will highlight that section of the command.



= Expression

YEARS(DateofBirth, <end_date>)

Figure 7.152: End Date Parameter

14. Click on the functions button f(x).
15. Select the **System Functions** option.

16. Click the **SYSTEMDATE** function. Once completed, the syntax will be inserted on the dialog window.

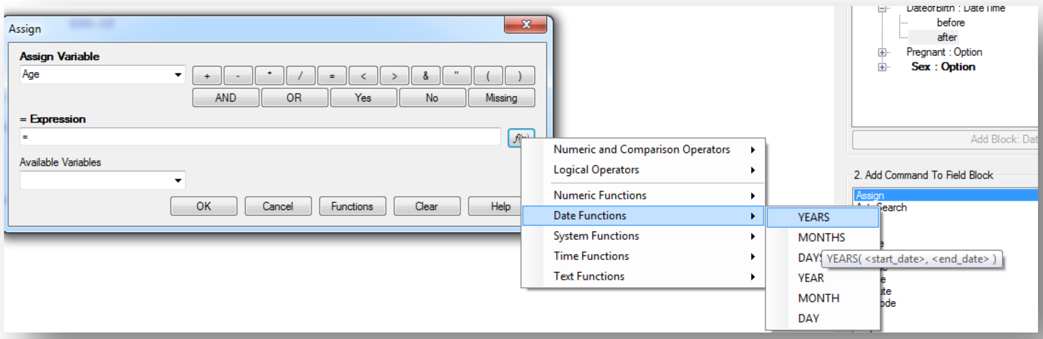
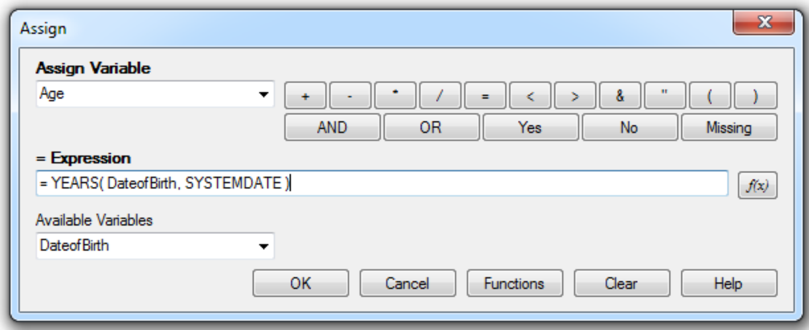


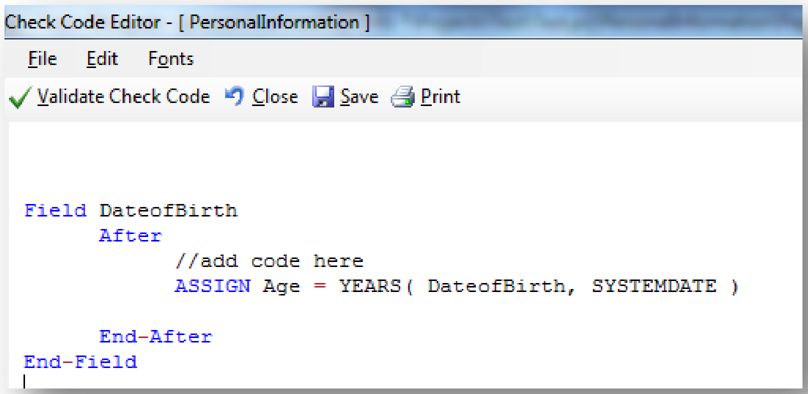
Figure 7.153: Assign Date Function options



```
= YEARS( DateofBirth, SYSTEMDATE )
```

Figure 7.154: Assign dialog box

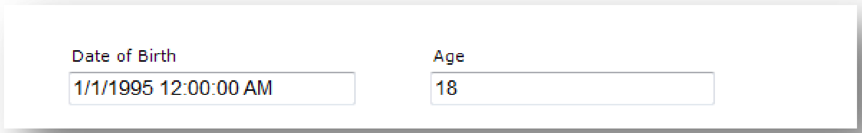
17. Click **OK**. Check Code will appear in the Check Code Editor.
18. Click the **Verify Check Code** button in the Check Code Editor.
19. Click the **Save** button in the Check Code Editor.



```
Field DateofBirth
      After
            //add code here
            ASSIGN Age = YEARS( DateofBirth, SYSTEMDATE )

      End-After
End-Field
```

Figure 7.155: Assign Check Code Command

When **Date of Birth** is entered into the form, the **Age** field will automatically populate.



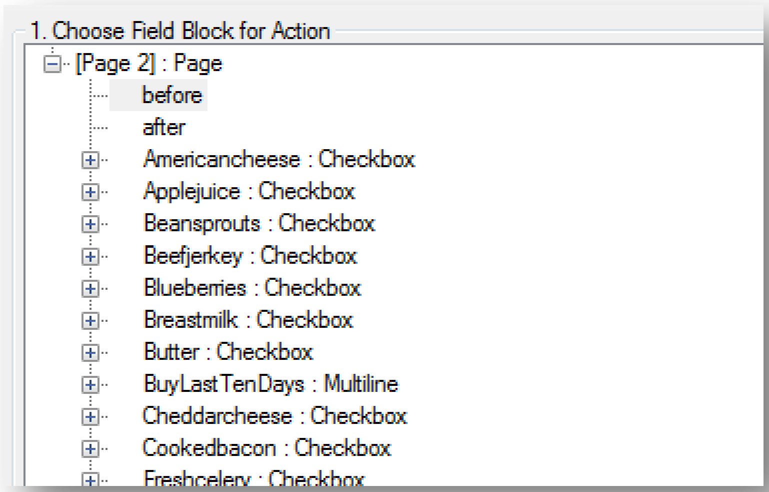
Date of Birth	Age
1/1/1995 12:00:00 AM	18

Figure 7.156: Assignment of Age in Enter Data

Interact with Users with the DIALOG command

The **DIALOG** command provides interaction with data entry personnel from within the program. Dialogs can display information, ask for and receive input, and offer lists to make choices. In the following example, the **DIALOG** command creates a reminder that all fields on page two of the survey must be completed.

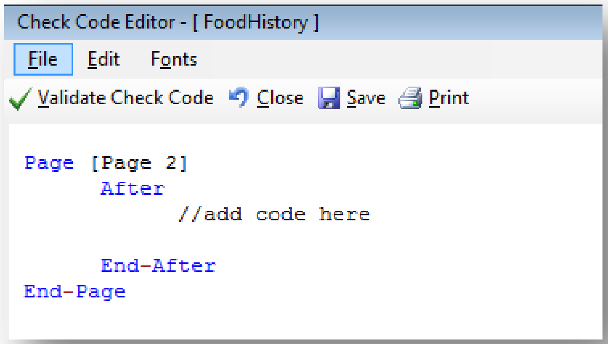
1. From Form Designer, Open the **FoodHistory_NoCheckCode** form in the **EColi.prj**.
2. Click **Check Code** or select **Tools > Check Code Editor**. The Check Code Editor opens.
3. From the **Choose Field Block for Action** list box, select **Page 2**. The action should occur before the page is loaded.
4. Select **Before** from the Before or After Section.
5. Click the **Add Block** button.


1. Choose Field Block for Action

- [Page 2] : Page
  - before
  - after
  - Americancheese : Checkbox
  - Applejuice : Checkbox
  - Beansprouts : Checkbox
  - Beefjerkey : Checkbox
  - Blueberries : Checkbox
  - Breastmilk : Checkbox
  - Butter : Checkbox
  - BuyLastTenDays : Multiline
  - Cheddarcheese : Checkbox
  - Cookedbacon : Checkbox
  - Freshcelery : Checkbox

Figure 7.157: Dialog Block for Action

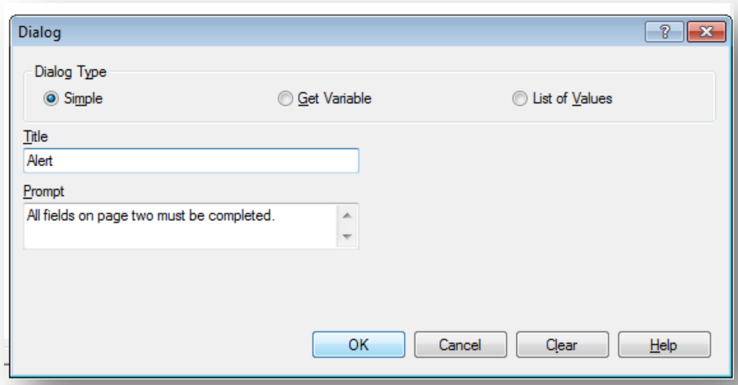
6. The code appears in the Check Code Editor.



```
Page [Page 2]
       After
              //add code here

       End-After
End-Page
```

Figure 7.158: Dialog Block Code

7. From the **Add Command to Field Block** list box, select **Dialog**. The **DIALOG** box opens.
8. Select **Simple** from the **Dialog Type** radio button options.
9. In the **Title** field, type ***Alert***.
10. In the **Promp**t field, type ***All fields on page two must be completed***.



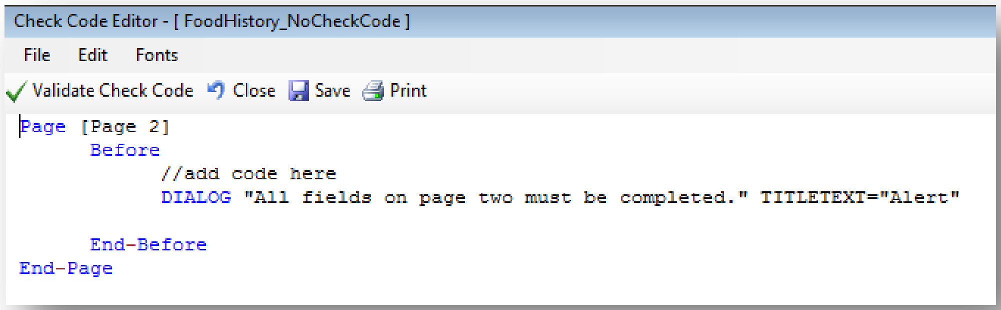
Dialog Type: [✓ Simple]

Title: Alert

Prompt: All fields on page two must be completed.

Figure 7.159: Dialog command box

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



```
Page [Page 2]
        Before
                //add code here
                DIALOG "All fields on page two must be completed." TITLETEXT="Alert"

        End-Before
End-Page
```

Figure 7.160: Dialog Check Code Command

12. Click the **Verify Check Code** button from the Check Code Editor.
13. Click the **Save** button in the Check Code Editor.

Searching for Records – Using the AUTOSEARCH command

The Autosearch command automatically searches for an existing record that matches the entered values and notifies the user. A choice is then offered between editing the matching record or continuing with data entry in the new record. Duplicate records are detected and may be prevented with the AutoSearch command. For this example, we will incorporate the Autosearch command on the **CaseID** field.

1. From Form Designer, Open the **FoodHistory_NoCheckCode** form in the **EColi.prj**.
2. Click **Check Code** or select **Tools > Check Code Editor**. The Check Code Editor opens.

3. From the **Choose Field Block for Action section,** expand the node for the page where the **CaseID** field is located, which in this example is Page 1.
4. Expand the node for the field **CaseID**.
5. Double click on the **after** event.
6. A block of code for the **CaseID** field is created and displayed in the Check Code Editor.

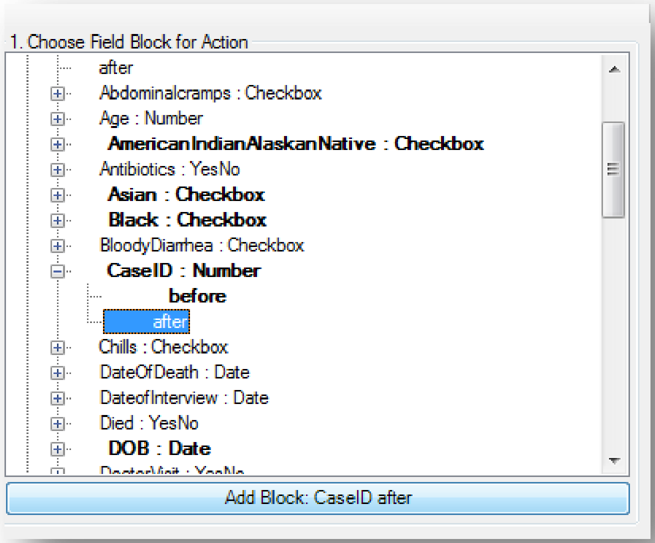
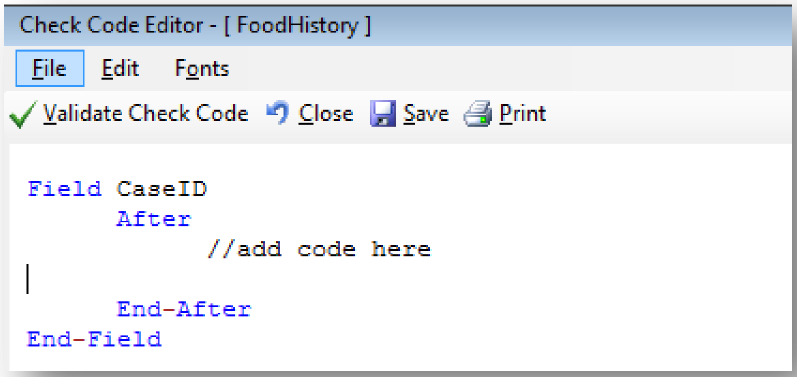


Figure 7.161: AutoSearch Block for Action

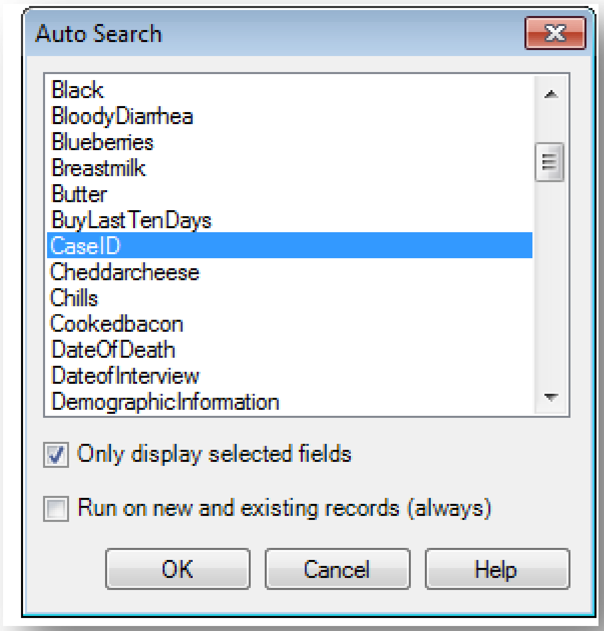
7. The code appears in the Check Code Editor.



```
Field CaseID
      After
            //add code here

      End-After
End-Field
```

Figure 7.162: AutoSearch Block Code

8. From the **Add Command to Field Block** list box, click **Autosearch**. The Autosearch window opens.
9. Select the variable(s) to be searched during data entry. In this example, select **CaseID**.

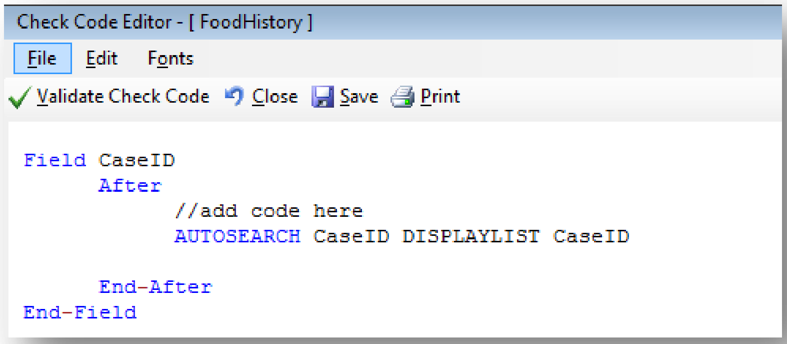


[✓] Only display selected fields

[ ] Run on new and existing records (always)

Figure 7.163: Auto Search dialog box

10. Click **OK**. The code appears in the Check Code Editor window.



```
Field CaseID
      After
            //add code here
            AUTOSEARCH CaseID DISPLAYLIST CaseID

      End-After
End-Field
```

Figure 7.164: AutoSearch Check Code Command

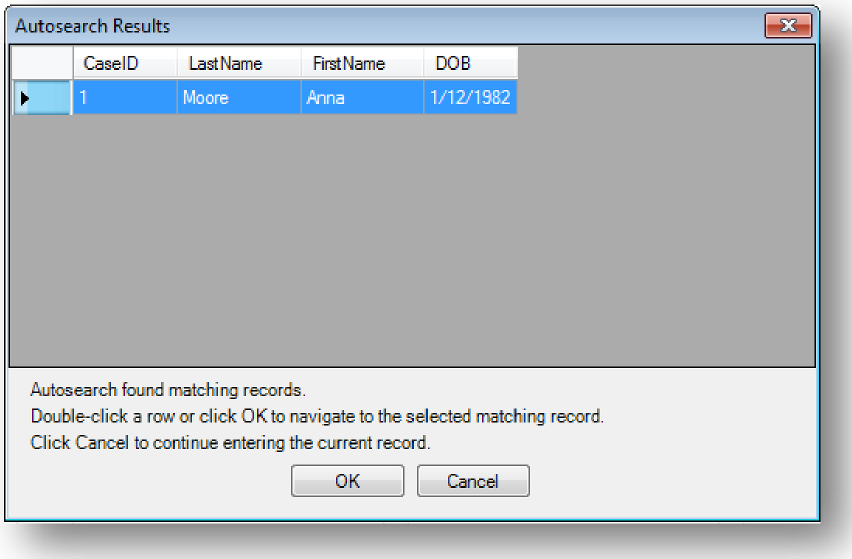
11. Click the **Verify Check Code** button from the Check Code Editor.
12. Click the **Save** button from the Check Code Editor.

13. Click **Close** to return to the form.

When a duplicate record is entered in the Enter Data tool, the **Autosearch** dialog box opens with all the matching records listed. To view the potential duplicate record, double-click the arrow that appears next to the record. The field where the potential duplicate was entered is cleared. Alternatively, click **Cancel** to remain on the current record and accept the duplicate value. To display other variables when a matching record is found, add the variable names after the **DISPLAYLIST** parameter (i.e. Last Name, First Name and Date of Birth as shown below).

```
Field CaseID
After
//add code here
AUTOSEARCH CaseId DISPLAYLIST CaseID LastName FirstName DOB
End-After
End-Field
```

In the check code above, the autosearch will be done on the field **CaseID** and if any matching records are detected, the fields **CaseID, LastName, FirstName** and **DOB** will be displayed on the grid.



	CaseID	LastName	FirstName	DOB
▶	1	Moore	Anna	1/12/1982

Autosearch found matching records.
Double-click a row or click OK to navigate to the selected matching record.
Click Cancel to continue entering the current record.

Figure 7.165: Autosearch Results

Note: For more information on using Autosearch, please see the Autosearch topic in the Command Reference.

Copy the Value of a Field from a Main Form to a Related Form

When users have developed a relational database setup using Epi Info™ 7, it might be appropriate to transfer values entered in the parent form (i.e. core demographics) into the child form (i.e. Visits information). In order to accomplish this process, Check Code must be created for a value from a field in the main form to appear in a related form (i.e., there may be a Case ID Number or Patient's Name that needs to be visible in the parent and child forms).

The following instructions assume the parent and child forms already exist. Let's assume that the Parent form is called **Surveillance** while the Child form is called **Hepatitis.** The field to be copied needs to exist in the parent form or be created in the parent form prior to the incorporation of the check code in the child form (i.e. Last Name in parent form will be passed to Last Name in child form). Let's make the assumption that the name of the field on the Parent form whose value will be copied to the Child form is called **PatientId**.

1. From Form Designer, open your project and click on the child form name from the Project Explorer tree.
2. Create a new field. The new field must be the same field type as the field being copied from the parent form. For this example, use **PatientId**.
3. Select the **Read Only** option.
4. Click **OK**. The new field appears in the form. This is where the value from the parent form will be assigned and displayed during data entry on the child form.
5. Click **Check Code** or select **Tools > Check Code Editor**. The Check Code Editor opens.
6. From the **Choose Field Block for Action** list box, select the page corresponding to where the PatientId field was placed. Let's make the assumption that this field was created on Page 1. For the Page, select the **before** from the Before or After Section.
7. Click the **Add Block** button.

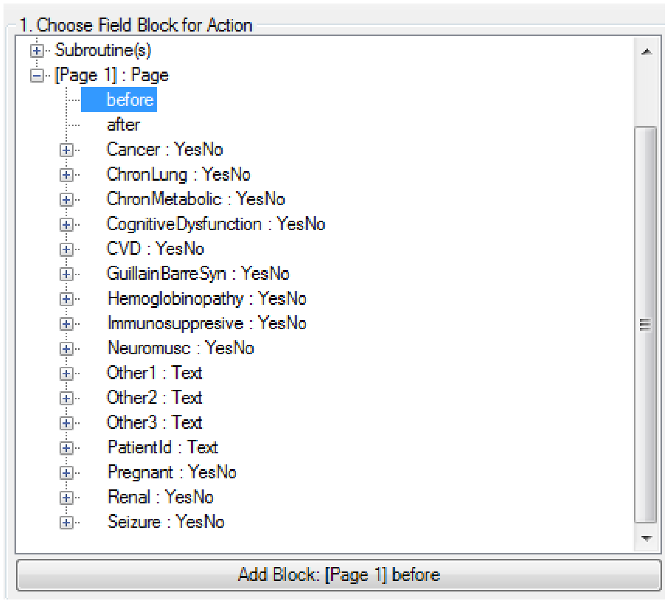


Figure 7.166: Choose Field Block for Action

8. From the **Add Command to Field Block list** box, click **Assign**. The **ASSIGN** dialog box appears.
9. From the **Assign Variable** drop-down list, select the new variable, **PatientId**.
10. In the = Expression area, type the **field name** from the parent form, in this case **PatientId**. This field name must be prefixed by the parent form's name followed by a period in the Assign expression. We will need to make that modification once the command is written into the Program Editor.

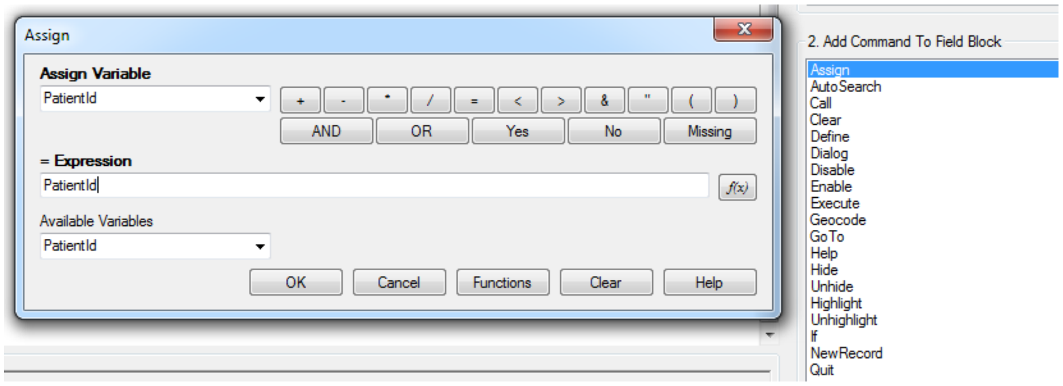
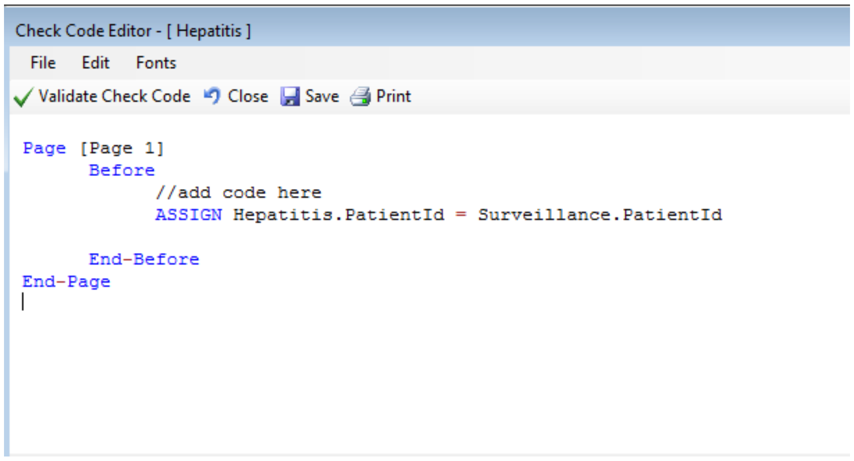

Assign
Assign Variable: PatientId
= Expression: PatientId
Available Variables: PatientId

Figure 7.167: Copy Value- Assign dialog box

11. Click **OK**.

12. The code appears in the Check Code Editor. If the child form already has a field with the same name as the one being copied from the parent form, it is important to distinguish the parent's field name. This field name must be prefixed by the parent form's name followed by a period in the Assign expression. In this example, Hepatitis is the name of the child form while Surveillance is the name of the parent form. If the child form does not have a field with the same name as the one being copied from the parent form, it is sufficient to indicate just the field name. Therefore, for this example, this modification will be required to the syntax since the field is called the same on both forms. Once completed, your syntax should look like the one on Figure 3.32.



```
Page [Page 1]
      Before
            //add code here
            ASSIGN Hepatitis.PatientId = Surveillance.PatientId

      End-Before
End-Page
```

Figure 7.168: Copy Value Check Code Command

13. Click the **Validate Check Code** button and, if needed, correct any issues.
14. Click **Save.**

Concatenate Fields

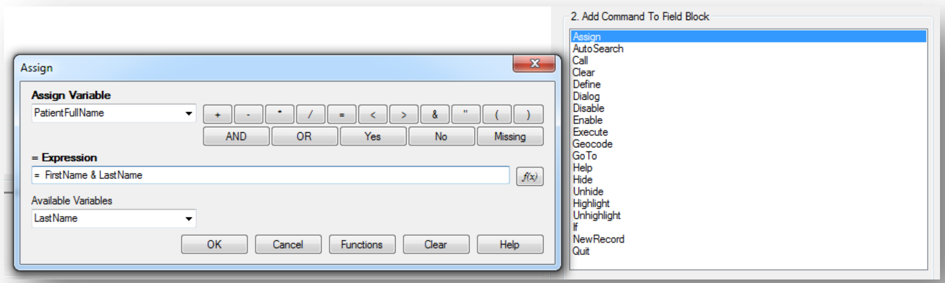
When writing check code for concatenating fields, syntax will be executed for new records entered. The code will not go back and populate previously entered data. If previously entered data need to be concatenated, use concatenation commands from the Classic Analysis section of the manual.

Concatenate Fields with the Ampersand '&' Operator

This example illustrates how to join data from two fields and assign it into a third field using the '&' operator. In this example, **Patient Full Name** will be assigned to the concatenation of **First Name** and **Last Name**.

1. In Form Designer, open your form.
2. Click **Check Code**. The Check Code Editor opens.
3. From the **Choose Field Block for Action** list box, select **LastName**.
4. Select **after** from the Before or After Section.
5. Click the **Add Block** button.
6. From the **Add Command to Field Block list** box, click **Assign**. The **ASSIGN** dialog opens.

7. From the **Assign Variable** drop-down list, select the field to contain the concatenated value.
8. Create the = Expression using the '&' operator. In this example, **ASSIGN PatientFullName = FirstName & LastName**.


Assign

Assign Variable: PatientFullName

= Expression: = FirstName & LastName

Available Variables: LastName

Figure 7.169: Concatenate- Assign dialog box

9. Click **OK**. Check Code appears in the Check Code Editor.

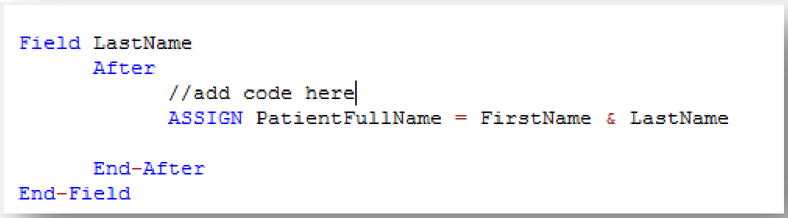
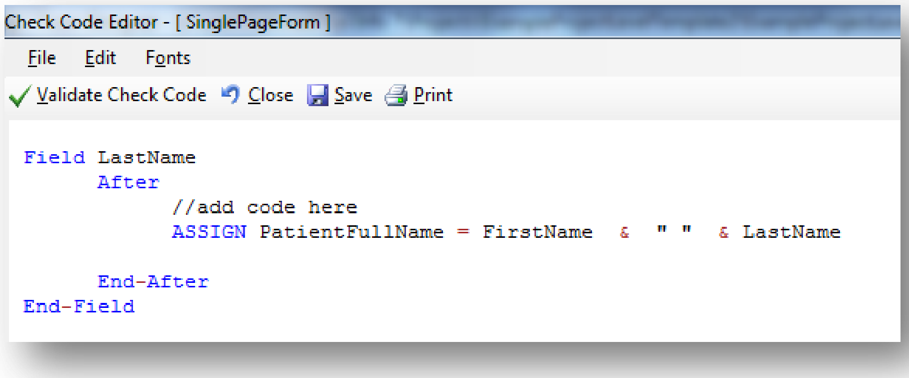

```
Field LastName
      After
            //add code here
            ASSIGN PatientFullName = FirstName & LastName

      End-After
End-Field
```

Figure 7.170: Concatenate No Space Check Code Command

10. Click the **Validate Check Code** button and correct any issues.
11. Click **Save.**

If in the Enter tool, **Carl** was entered for FirstName and **Gao** was entered for LastName, the result of PatientFullName would be **CarlGao**. There are no spaces between the names. To add a space between the names, simply modify the ASSIGN statement by adding a blank space in quotes between the first and last names as in the following statement:


Check Code Editor - [SinglePageForm]
File Edit Fonts
Validate Check Code Close Save Print

```
Field LastName
      After
            //add code here
            ASSIGN PatientFullName = FirstName  &  " "  & LastName

      End-After
End-Field
```

Figure 7.171: Concatenate Include Space Check Code Command

Concatenate Fields with the Substring Function

This example illustrates how to join parts of two variables to create a unique text ID. In this example, you will create a Patient ID made up of parts of the patient's last and first name. The ampersand (&) operator is used to join the two parts together.

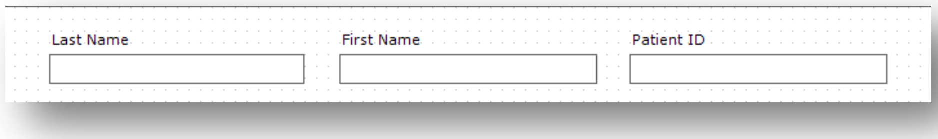


Figure 7.172: Enter Form - Name

1. From the Form Designer, click **Check Code**. The Check Code Editor opens.
2. From the **Choose Field Block for Action** list box, select **FirstName**.
3. Select **after** from the Before or After Section.
4. Click the **Add Block** button.
5. From the **Add Command to Field Block** list box click **Assign**. The **ASSIGN** dialog box opens.
6. From the **Assign Variable** drop-down list, select the field to contain the concatenated value. In this example, select **PatientID**.
7. Create the =Expression using the SUBSTRING syntax.

- SUBSTRING(<variable>, position #, #characters)
- <variable> is the field or variable
- position # is the position of the first character to be extracted from the variable
- #characters is the number of characters to extract

In this example, the **PatientID** variable contains a combination of the first position and four characters of the last name plus the first position and three characters of the first name.

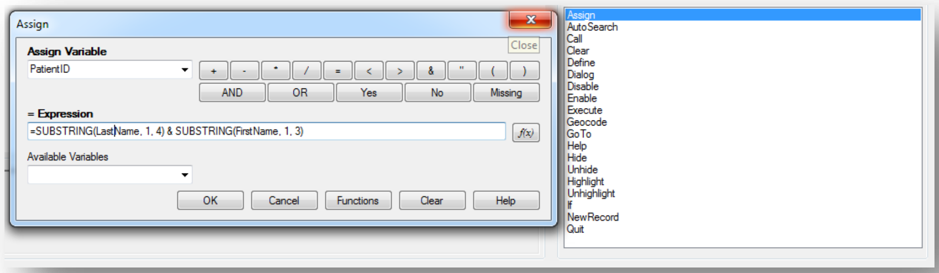
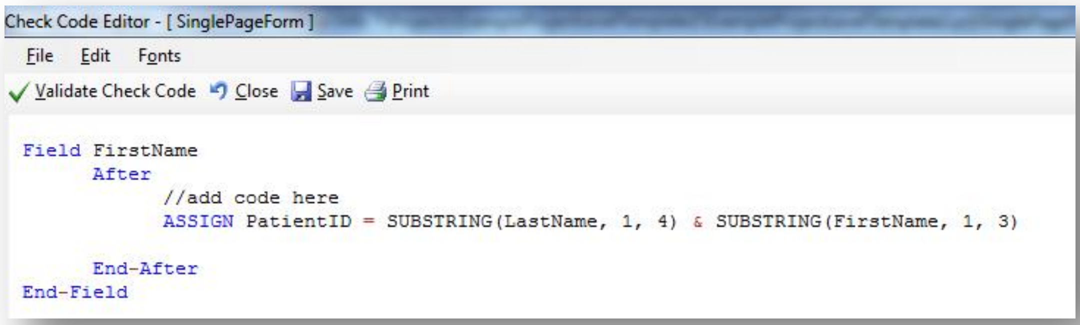


Figure 7.173: Concatenate with Substring Assign dialog box

8. Click **OK**. The code appears in the Check Code Editor window.


Check Code Editor - [SinglePageForm]
File Edit Fonts
Validate Check Code Close Save Print

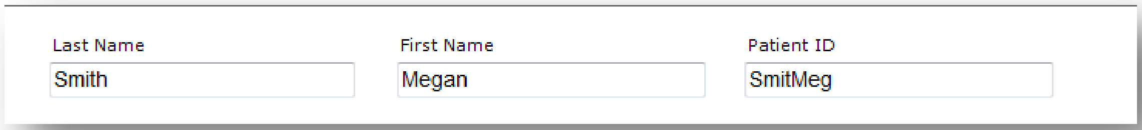
```
Field FirstName
      After
            //add code here
            ASSIGN PatientID = SUBSTRING(LastName, 1, 4) & SUBSTRING(FirstName, 1, 3)

      End-After
End-Field
```

Figure 7.174: Concatenate with Substring Check Code Command

9. Click the **Validate Check Code** button and correct any issues.
10. Click the **Save** button.

The example functions as such:

- Last Name: Smith
- First Name: Megan
- Patient ID: SmitMeg
- The **Patient ID** field being calculated is Read Only.



Last Name	First Name	Patient ID
Smith	Megan	SmitMeg

Figure 7.175: Concatenate with Substring Enter Form

Create Check Code for Option Box Fields

Check Code can be added to option box variables. Code can be added to any line/choice present in the option boxes. Use the Check Code Editor to create complex Check Code for option box variables.

For this example, the check code created uses the **GOTO** command. Check Code was used to create the following scenario. If the answer to *TestOptions* is Choice 1, the cursor will jump to *Question 2*. If the answer to *TestOptions* is *Choice 2* or *Choice 3*, the cursor will jump to *Question 1*. Check the tab order before creating the Check Code to ensure that the tab order is correct.

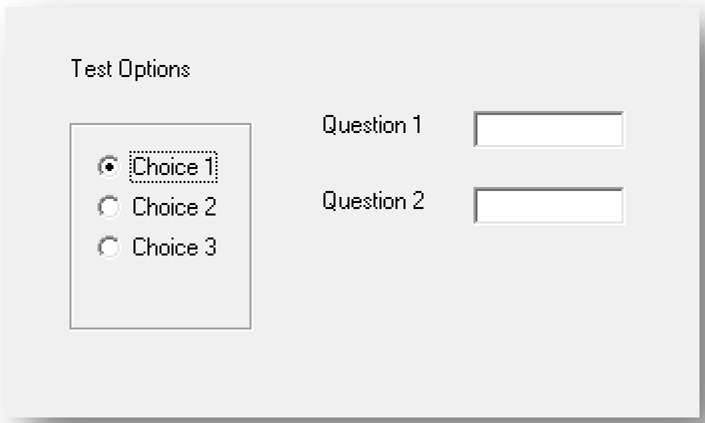
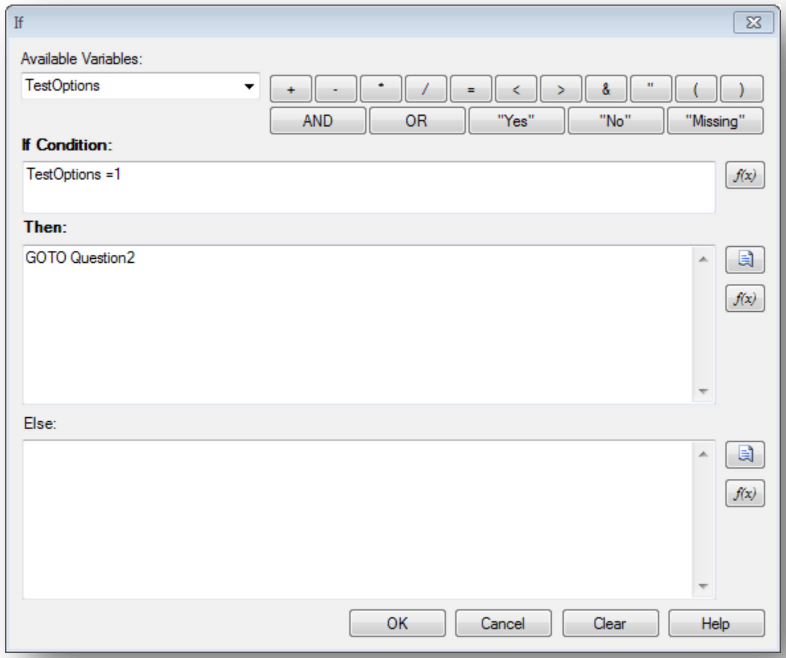


Figure 7.176: Text Options box

1. Create an Option Box in a form. Note the name of the variable.
   - The variable is named TestOptions.
   - Each text line that represents a choice in the form represents a numeric position in the Check Code Editor.
   - For example, there are three lines/choices that can be made in the variable *TestOptions*. In the Check Code Editor the choices are numeric, choice 1 = position 0, choice 2 = position 1, and choice 3 = position 2.
2. Open the **Check Code Editor**.
3. From the Choose Field Block for Action list box, select **TestOptions**.
4. Select **after** from the Before or After Section.
5. Click the **Add Block** button.
6. From the **Add Command to Field Block** list box, click **If**. The **IF** dialog box opens.
7. From the **If Condition** field, type **Test Options = 1**.
8. Remember that the number 1 in this instance represents a text value called *Choice 2*.
9. Click the **Code Snippet** button in the **Then** section. A list of available commands appears.
10. From the command list, select **GoTo**. The **GOTO** dialog box opens.

11. Select **Question2**.
12. Click **OK**.



TestOptions =1

GOTO Question2

Figure 7.177: If dialog box

13. Click **Validate Check Code** and correct any issues.
14. Click **Save.**

Delete a Line of Code from the Check Code Editor

To delete a line(s) of check code from the Check Code Editor, you can complete the following steps:

1. Highlight the line(s) of code/text that you desire to delete**.**
2. Press the **Delete** key on your keyboard.
3. Once done, from the Check Code Editor toolbar, click **Save**.

Note: Be sure of all deletions made. No confirmation prompt or undo button will appear prior to deletion.

Disable

If there isn't a need to capture information for a particular field, then it can be disabled. Disable is usually used with the IF, THEN, ELSE conditions. In this example, the field DoctorVisitDate will be disabled if the response to the DoctorVisit is *No*.

1. Open the **FoodHistory_NoCheckCode** form in the **EColi.prj.**
2. Click **Check Code**. The Check Code Editor opens.
3. Select the **DoctorVisit** from the **Choose Field Block for Action** list box.
4. Select **after** from the Before or After Section.
5. Click the **Add Block** button.

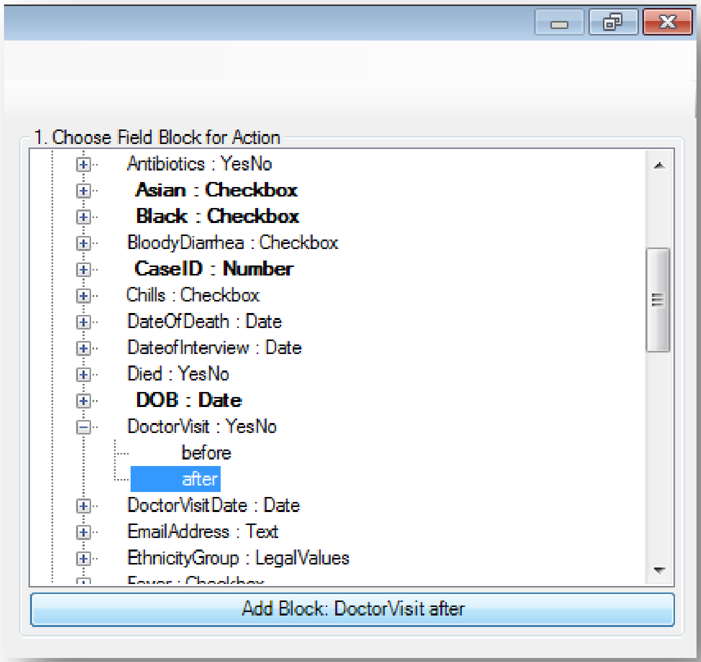
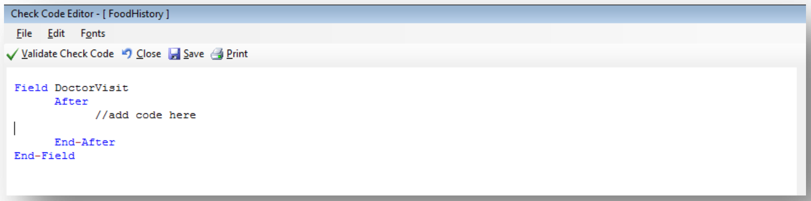


Figure 7.178: Disable Block for Action

6. The code block appears in the Check Code Editor.

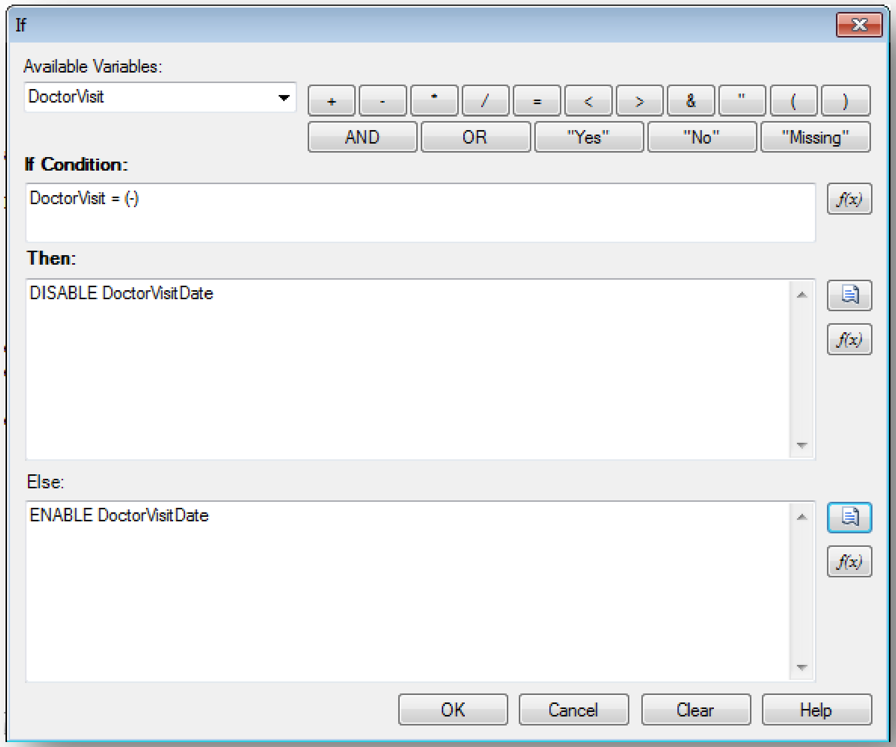


```
Field DoctorVisit
      After
            //add code here
|
      End-After
End-Field
```

Figure 7.179: Disable Block Command

7. Click **IF** from the **Add Command to Field Block** list box. The **IF** dialog box opens.
8. From the **Available Variables** drop-down list, select the **field** to contain the action. For this example, select **DoctorVisit**. The selected variable appears in the **If Condition** field.
9. From the Operators, click **=**.
10. From the Operators, click **No**. The **If Condition** field will read DoctorVisit=(-).

11. Click the **Code Snippet** button in the **Then** section. A list of available commands appears.
12. From the command list, select **DISABLE**. The **DISABLE** dialog box opens.
13. Select the **field** to be disabled based on a **No** answer from the list of variables. For this example, select **DoctorVisitDate.**
14. Click **OK** to return to the **IF** dialog box.
15. Click the **Code Snippet** button in the **Else** section. A list of available commands appears.
16. From the command list, select **Enable**. The **Enable** dialog box opens.
17. Select the field to enable based on a **Yes** answer from the list of variables. For this example, select **DoctorVisitDate.**
18. In the **Enable** dialog box, click **OK** to return to the **If** dialog box.



Available Variables: DoctorVisit

If Condition:

```
DoctorVisit = (-)
```

Then:

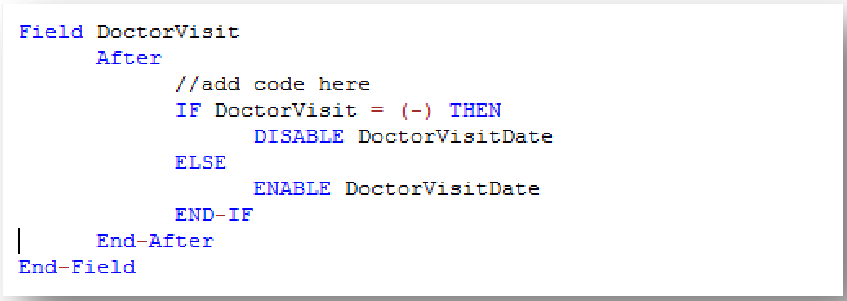
```
DISABLE DoctorVisitDate
```

Else:

```
ENABLE DoctorVisitDate
```

Figure 7.180: Disable- If dialog box

19. Click **OK**. The code appears in the Check Code Editor. The example code appears as:



```
Field DoctorVisit
      After
            //add code here
            IF DoctorVisit = (-) THEN
                  DISABLE DoctorVisitDate
            ELSE
                  ENABLE DoctorVisitDate
            END-IF
      End-After
End-Field
```

Figure 7.181: Disable Check Code Command

20. Click Verify Check Code button.
21. Click **Save**.
22. Click **Close** to return to the form.

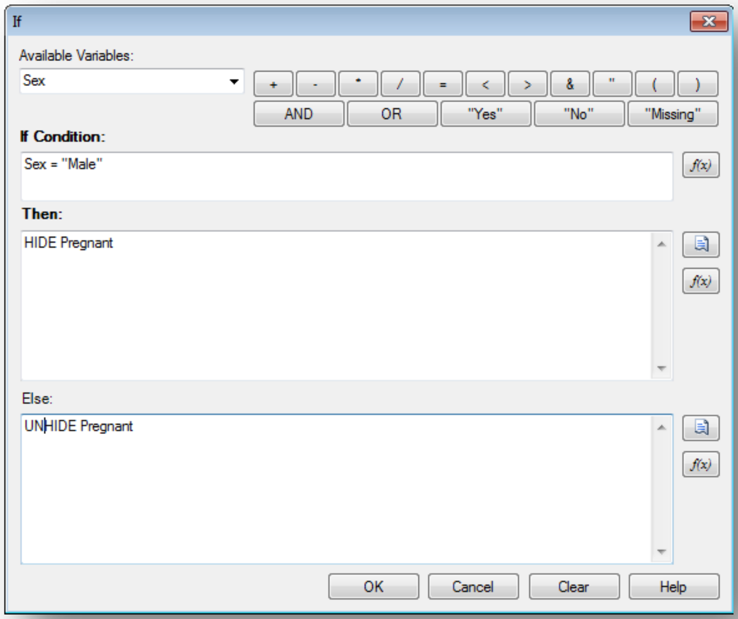
To test the Disable, open the form in the Enter Data tool. When **No** is entered for **DoctorVisit**, **DoctorVisitDate** should become disabled.

Hide

A field may be hidden if there is not a need to capture information for a particular field or if it does not apply based on previously answered questions. The field on the form would be hidden from the form. In the following example, the Pregnant option box will be hidden based on the users Sex; Female or Male.

1. Click **Check Code**. The Check Code Editor opens.
2. Select **Sex** 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 data is entered and accepted.
5. The code appears in the **Check Code Editor**.
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 **field** to contain the action. For this example, select **Sex**. The selected variable appears in the **If Condition** field.
8. From the Operators, click **=**.
9. From the **If Condition** field, type **Sex = "Male"**. Remember that Sex is a text field and the value must be enclosed in quotes.
10. Click the **Code Snippet** button in the **Then** section. A list of available commands appears.
11. From the command list, select **Hide**. The **Hide** dialog box opens.
12. Select **Pregnant**.

13. Click **OK.**



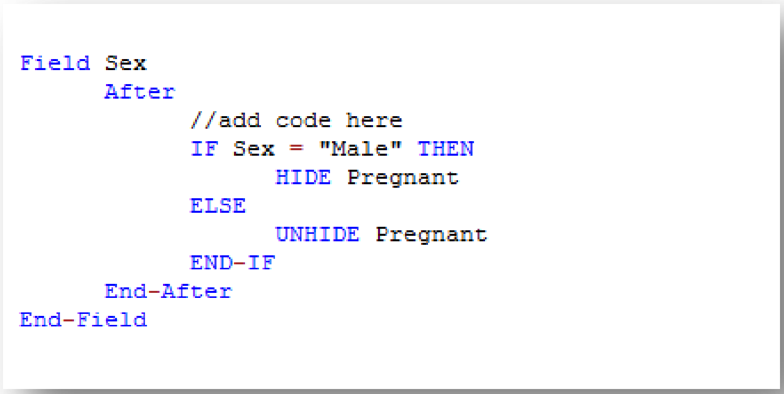
```
Sex = "Male"
```

```
HIDE Pregnant
```

```
UNHIDE Pregnant
```

Figure 7.182: Hide- 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 **UNHIDE**. The **Unhide** dialog box opens.
16. Select the field to enable based on a **Yes** answer from the list of variables. For this example, select **Pregnant.**
17. In the **Unhide** dialog box, click **OK** to return to the **If** dialog box.
18. Click **OK**.
19. Click **OK**. The code appears in the Check Code Editor.



```
Field Sex
      After
            //add code here
            IF Sex = "Male" THEN
                  HIDE Pregnant
            ELSE
                  UNHIDE Pregnant
            END-IF
      End-After
End-Field
```

Figure 7.183: Hide Check Code Command

20. Click the **Save** button.
21. Click **Close** to return to the form.

Note: The Unhide command will display any hidden fields. You can select this command in the Add Command to Field Block section of the Check Code Editor. It is recommended to also incorporate a CLEAR command with the HIDE command in order to set null any information previously entered into the Pregnant field.

Geocode

The Geocode command uses the text entered into Address to retrieve and populate Latitude and Longitude coordinates into your form.

1. From Form Designer, Open the **FoodHistory_NoCheckCode** form in the **EColi.prj**.
2. Click **Check Code**. The Check Code Editor opens.
3. Select **GetCoordinates** from the **Choose Field Block for Action** list box.
4. Select **Click.**
5. Click the **Add Block** button. This creates code to run after the **Get Coordinates** command button is clicked.

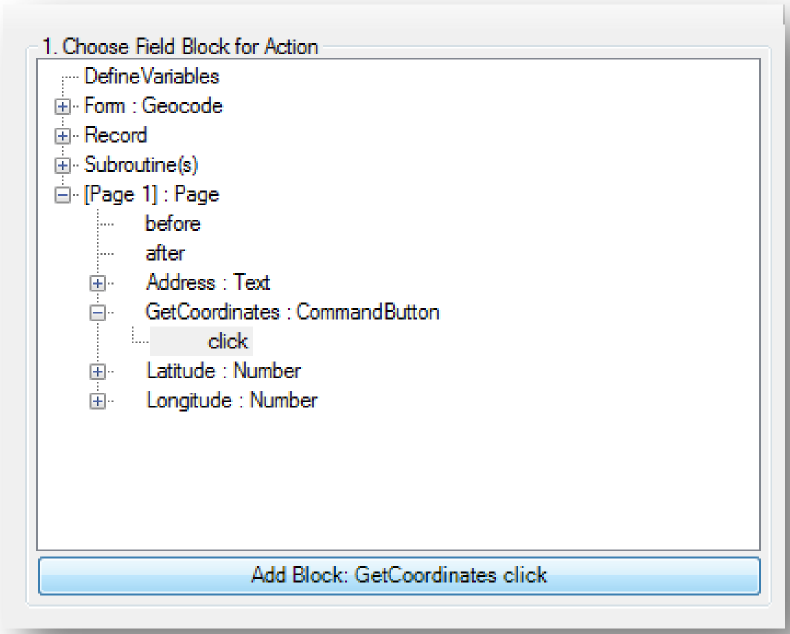


Figure 7.184: Geocode Block for Action

6. The code appears in the Check Code Editor.
7. Select **Geocode** from the **Add Command to Field Block** list box. The **Geocode** dialog box opens.

8. From the **Address** drop-down list, select **Address**.
9. From the **Latitude** drop-down list, select **Latitude**.
10. From the **Longitude** drop-down list, select **Longitude**.

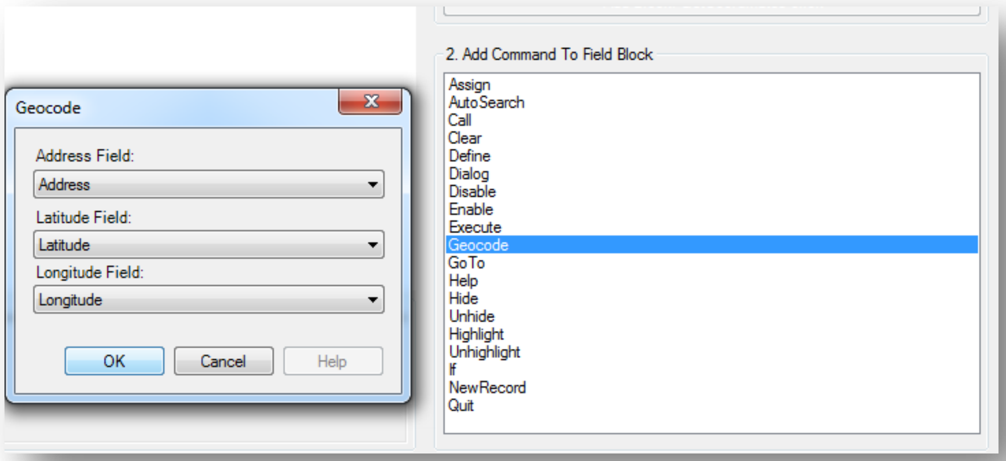
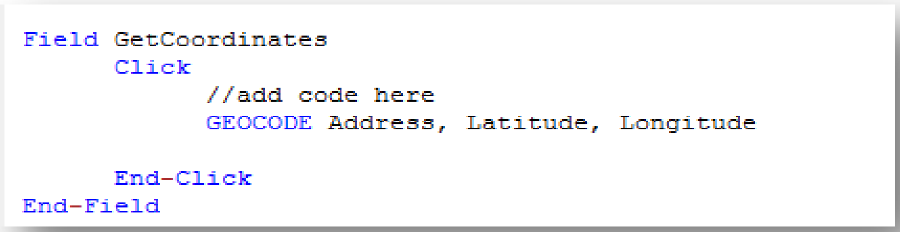


Figure 7.185: Geocode dialog box

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

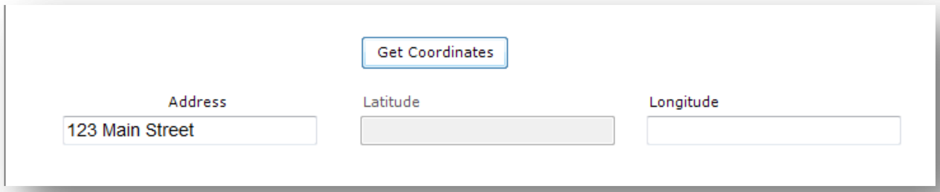


```
Field GetCoordinates
      Click
            //add code here
            GEOCODE Address, Latitude, Longitude

      End-Click
End-Field
```

Figure 7.186: Geocode Check Code Command

13. Click the **Save** button.
14. Click **Close** to return to the form.

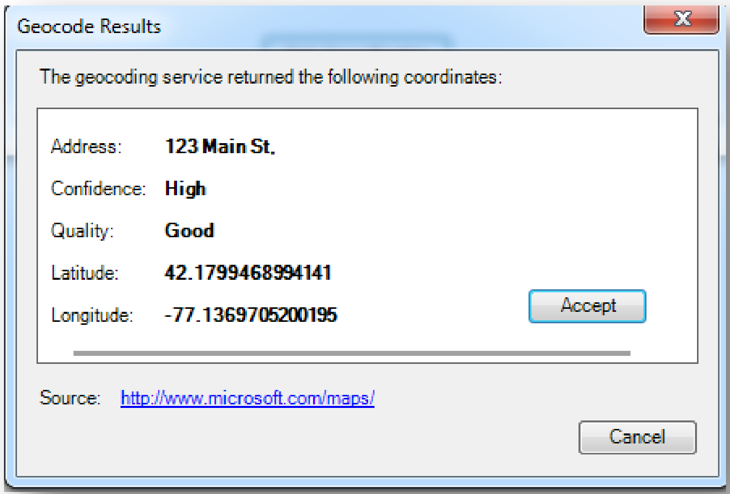
When the **Get Coordinates** command button is clicked after an address is entered into your form in the Enter Data tool, the **Geocode Results** dialog will appear.



123 Main Street

Figure 7.187: Geocode Enter Data Before

The **Geocode Results** dialog box contains, the Address entered into the form along with the Latitude and Longitude coordinates of that address. The confidence and quality of the geocoding service coordinates are also displayed.



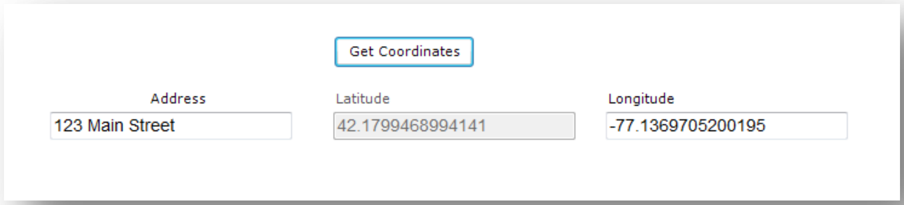
The geocoding service returned the following coordinates:

Address:	123 Main St.
Confidence:	High
Quality:	Good
Latitude:	42.1799468994141
Longitude:	-77.1369705200195

Source: http://www.microsoft.com/maps/

Figure 7.188: Geocode Results

After clicking **Accept** in the **Geocode Results** dialog box, the coordinates provided by the geocoding service are copied into the **Latitude** and **Longitude** fields in your form.



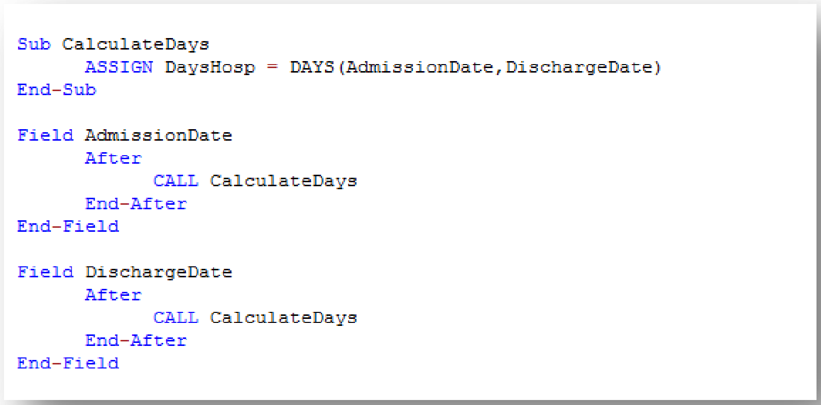
Address	Latitude	Longitude
123 Main Street	42.1799468994141	-77.1369705200195

Figure 7.189: Geocode Enter Data After

Additional Check Code Commands

Call

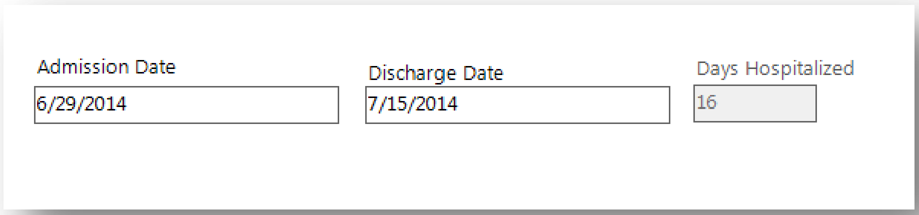
This command redirects to another command block in check code and returns after it has been executed. This command is commonly used with subroutines. Subroutines act as a unit of common check code, which is usually dependent on two variables. The benefit of using subroutines is that it allows maintenance of check code in one common location. Here is an example of a subroutine:



```
Sub CalculateDays
      ASSIGN DaysHosp = DAYS(AdmissionDate,DischargeDate)
End-Sub

Field AdmissionDate
      After
            CALL CalculateDays
      End-After
End-Field

Field DischargeDate
      After
            CALL CalculateDays
      End-After
End-Field
```

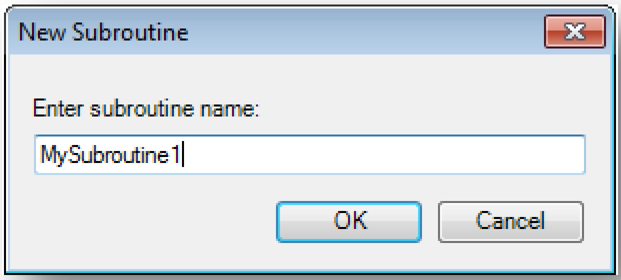
Figure 7.54: Check code syntax for Subroutines

In the example above, the calculation of days hospitalized would have been required to be placed in the AFTER event of the Admission Date and Discharge Date fields. However, through the usage of subroutines, users can update and maintain check code only in one location. By using the CALL command, the block of check code will execute when data are entered or updated in either field.



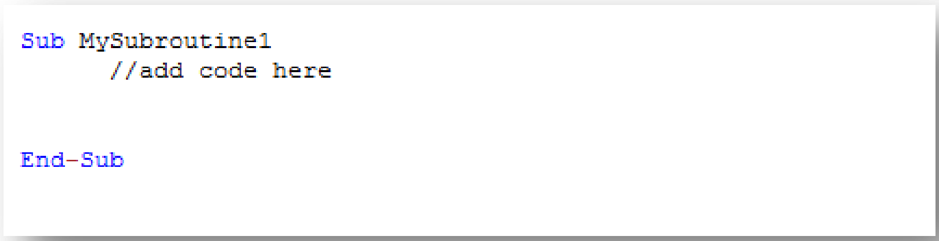
To create a subroutine, complete the following steps:

1. From Form Designer, Open your form.
2. Click **Check Code**. The Check Code Editor opens.
3. From the **Choose Field Block for Action** list box, expand the Subroutine item.
4. Double click on the **Add new item.** The *New Subroutine* window opens.
5. Assign a name to your subroutine (i.e. MySubroutine1).



1.

6. Click **OK**.

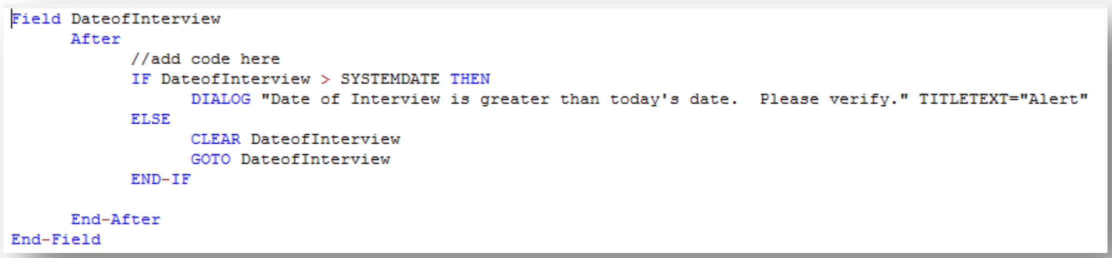


```
Sub MySubroutine1
      //add code here


End-Sub
```

A block of code for the subroutine called MySubroutine1 is created and displayed in the Check Code Editor. At this point, you can establish the check code commands to be incorporated into the subroutine. Make sure to Verify your code and Save once done.

Clear

CLEAR sets a field to a missing value, as though the field had been left blank. For example, it is useful to clear a previous entry in a field after an error has been detected. The CLEAR command can be followed by a GOTO command in order to place the cursor back into the field for further entry after an error. In the example below, the Date of Interview field is cleared after the user is notified that the date value entered is greater than today's date.



```
Field DateofInterview
      After
            //add code here
            IF DateofInterview > SYSTEMDATE THEN
                  DIALOG "Date of Interview is greater than today's date.  Please verify." TITLETEXT="Alert"
            ELSE
                  CLEAR DateofInterview
                  GOTO DateofInterview
            END-IF

      End-After
End-Field
```

Figure 7.55: Check code syntax for CLEAR command

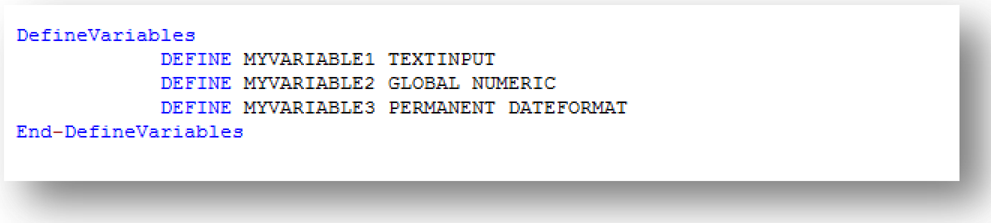
Define

This command creates a new variable. In Check Code, all user-defined variables are saved in the DEFINEVARIABLES section.

The proper syntax is:

DEFINE <variable name> {<scope>} {<field type indicator>}

- **<variable name>** represents the name of the variable to be created. <variable> cannot be a reserved word. For a list of reserved words, see the List of Reserved Words section of the User's Manual.
- **<scope>** is optional and is the level of visibility and availability of the new variable. This parameter must be one of the reserved words: STANDARD, GLOBAL, or PERMANENT. If omitted, STANDARD is assumed and a type indicator cannot be used.
- **<field type indicator>** is the data type of the new variable and must be one of the following reserved words: NUMERIC, TEXTINPUT, YN, DATEFORMAT, TIMEFORMAT and DLLOBJECT. If omitted, the variable type will be inferred based on the data type of the first value assigned to the variable. Thereafter, the variable type cannot be changed. An attempt to assign data of a different type to the variable will result in an error.



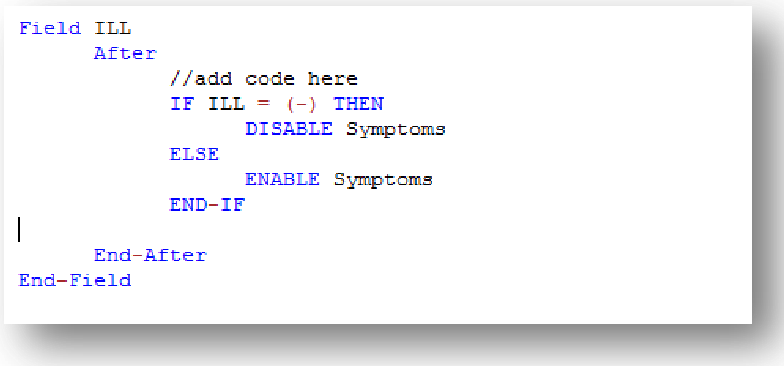
```
DefineVariables
            DEFINE MYVARIABLE1 TEXTINPUT
            DEFINE MYVARIABLE2 GLOBAL NUMERIC
            DEFINE MYVARIABLE3 PERMANENT DATEFORMAT
End-DefineVariables
```

Figure 7.56: Check code syntax for DEFINE command

Below is a description for the optional SCOPE parameter used with the DEFINE command.

- **STANDARD variables** retain their value only within the current record and are reset when a new record is loaded. Standard variables are used as temporary variables behaving like other fields in the database.
- **GLOBAL variables** retain values across related forms and when a new form is opened by the program, but are removed when the Enter program is closed. Global variables persist for the duration of program execution.
- **PERMANENT variables** are stored in the **EpiInfo.Config.xml** file and retain any value assigned until the value is changed by another assignment or the variable is undefined. Permanent variables are shared among Epi Info modules (Enter, Classic Analysis, etc.) and persist even if the computer is shut down.

Enable/Disable These two commands usually work in conjunction. The DISABLE command disallows data entry into a field while the ENABLE command allows data entry into a previously disabled field.



```
Field ILL
      After
            //add code here
            IF ILL = (-) THEN
                  DISABLE Symptoms
            ELSE
                  ENABLE Symptoms
            END-IF

      End-After
End-Field
```

Figure 7.57: Check code syntax for ENABLE/DISABLE command

Execute Use to execute a Windows program.
Executes a Windows program - either one explicitly named in the command or one designated within the Windows registry as appropriate for a document with the file extension that is named. This provides a mechanism for bringing up whatever program is the default on a computer without first knowing its name. The EXECUTE command accepts a series of paths, separated by semicolons, as in:

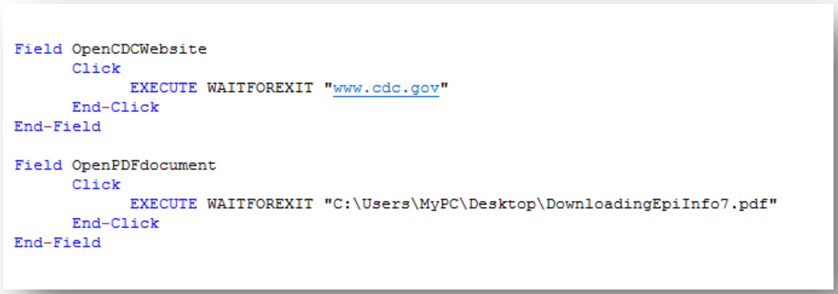
EXECUTE c:\Users\MyPC\myfile.xls;d:\myfile.xls

If the first is not found, the others are tried in succession. In Check Code, the EXECUTE command can be placed in any command block, but is often used with a command button. Check code syntax is executed when the user clicks on the command button.

EXECUTE WAITFOREXIT "<filename>"EXECUTE NOWAITFOREXIT "<filename>"

- The <filename> represents the path and program name for .exe (filename for registered Windows programs) and .com (filename -DOS binary executables) files.
- The <command-line parameters> represent any additional command-line arguments that the program can accept.
- When **Wait for Exit** command is specified (modal), the command should run and Enter should continue running. When **No Wait for Exit** command is specified (non-modal), Enter should wait until the executed program closes before continuing. When EXECUTE is run modally, permanent variables are written before the command is executed and reloaded after the command is executed.

If the example below, the CDC website page is opened when the user clicks on a command button called *OpenCDCWebsite*. A .pdf file is automatically opened when the user clicks on a command button called *OpenPDFdocument*.



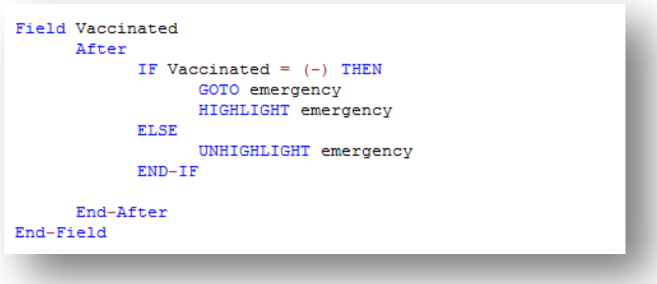
```
Field OpenCDCWebsite
      Click
            EXECUTE WAITFOREXIT "www.cdc.gov"
      End-Click
End-Field

Field OpenPDFdocument
      Click
            EXECUTE WAITFOREXIT "C:\Users\MyPC\Desktop\DownloadingEpiInfo7.pdf"
      End-Click
End-Field
```

Figure 7.58: Check code syntax for EXECUTE command

Help

The Help command allows you to pop up a help window containing a message, or even a window on a large file that allows the user to move from one block of text to another by choosing highlighted portions of the text. This is a simple form of what is known as "hypertext."

Highlight/Unhighlight This command emphasizes the location of a field by highlighting in bright yellow, for example, if a data entry error was detected. It also unhighlights the field if needed. In the example below, the field Emergency is highlighted if the response to the field Vaccinated = "No". All vaccinated questions are skipped and the field "Did you visit the emergency room?" is highlighted.

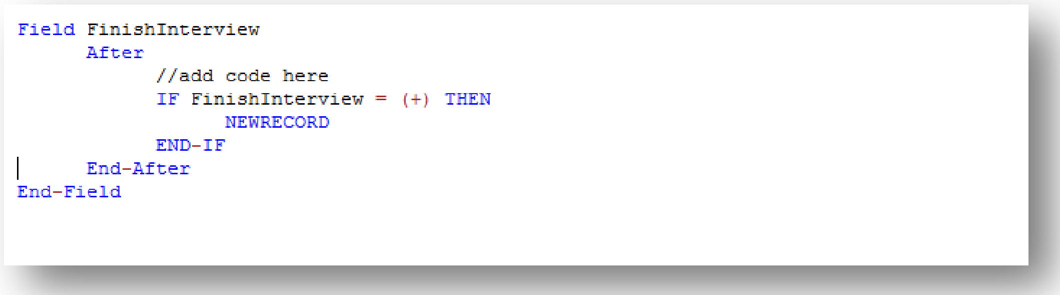


```
Field Vaccinated
      After
            IF Vaccinated = (-) THEN
                  GOTO emergency
                  HIGHLIGHT emergency
            ELSE
                  UNHIGHLIGHT emergency
            END-IF

      End-After
End-Field
```

Figure 7.59: Check code syntax for HIGHLIGHT/UNHIGHLIGHT command

New Record

This command saves the current records data and opens a new record for data entry.

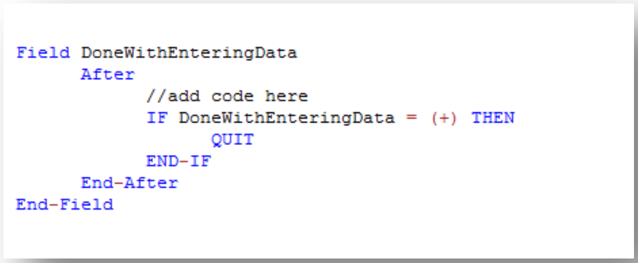


```
Field FinishInterview
      After
            //add code here
            IF FinishInterview = (+) THEN
                  NEWRECORD
            END-IF
      End-After
End-Field
```

Figure 7.60: Check code syntax for NEWRECORD command

Quit

This command allows the saving of the current record and closing of the Enter application.

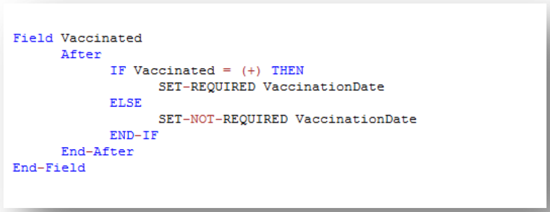


```
Field DoneWithEnteringData
      After
            //add code here
            IF DoneWithEnteringData = (+) THEN
                  QUIT
            END-IF
      End-After
End-Field
```

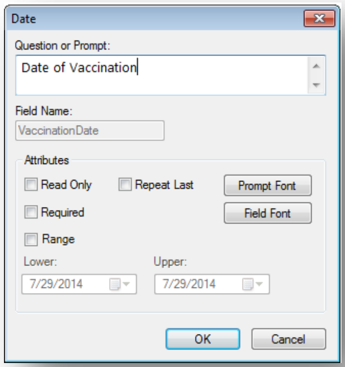
Figure 7.61: Check code syntax for QUIT command

Set-Required-Set Not Required-

Sometimes users might want to set up a field as required only if a specific criterion is met. As with the REQUIRED property, if a field is set to required during data entry through check code, the Enter module will not allow further page navigation until a value has been entered into the field. In the example below, the field VaccinationDate is set as required if the Vaccination question is answered as *Yes* using the **Set-Required** command. Notice that the REQUIRE property has not been set during the creation of the field but it has been set with the check code. Once the criterion is met, the field is set back to its original state by using the **Set-Not-Required** command.



```
Field Vaccinated
      After
            IF Vaccinated = (+) THEN
                  SET-REQUIRED VaccinationDate
            ELSE
                  SET-NOT-REQUIRED VaccinationDate
            END-IF
      End-After
End-Field
```



Date

Question or Prompt: Date of Vaccination

Field Name: VaccinationDate

Attributes

☐ Read Only ☐ Repeat Last

☐ Required

☐ Range

Lower: 7/29/2014 Upper: 7/29/2014

Figure 7.62: Check code syntax for SET REQUIRED command

Check Features Covered Elsewhere- The **Enter** tool can be programmed to do many interesting and complex operations not discussed in this chapter, including mathematical and logical operations with more than one field, popping up help windows, and calling programs in other languages that act on the contents of fields during data entry.

How to use EpiWeek Function

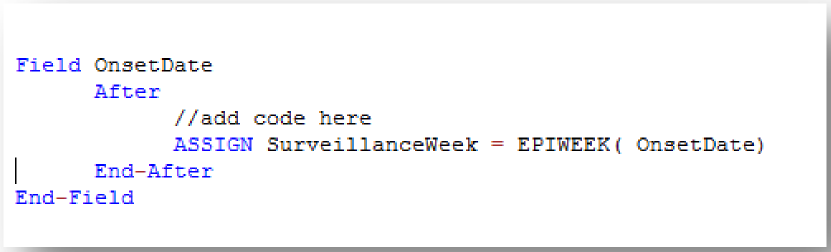
Epidemiological weeks are usually complete weeks. The ministries of health around the world define the day of the week as the first epidemiologic day. As a result, some countries may consider Sunday as the first day of the week while others may consider the first day of the week to be either Saturday or Monday.

By default, Epi Info™ 7 marks Sunday as the beginning of the epidemiological week. However, the parameter for the Epiweek function can be modified in order to change the beginning of the epidemiological week as desired.

If the year of occurrence is not relevant, use the EpiWeek method instead. The advantage of using EpiWeek is that the value is returned as a number and it can be stored in a numeric field. EpiWeek takes one required parameter that must be a date. The week is calculated relative to the year of the date provided. The Epidemiological week is calculated using the following code:

```
ASSIGN MyEpiWeek = EPIWEEK( <start_date>, {<first_day_of_week>} )
```

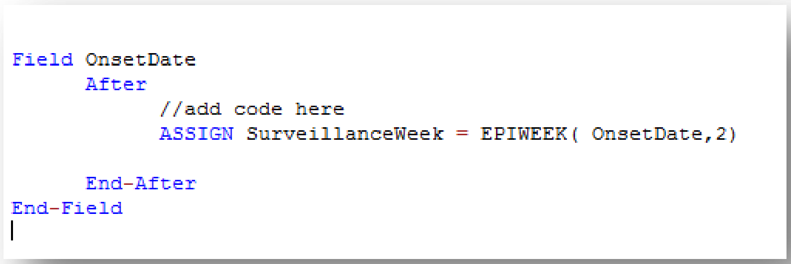
The example below shows the check code needed for automatically calculating the corresponding Epi Week into a field called *SurveillanceWeek* based on the value entered in a date field called *OnsetDate*.



```
Field OnsetDate
	After
		//add code here
		ASSIGN SurveillanceWeek = EPIWEEK( OnsetDate)
	End-After
End-Field
```

Figure 7.63: Check code syntax for EPIWEEK function

In the other hand, if you are required to modify the first day of the week parameter because the week does not start on a Sunday, the check code syntax would need to be updated. In the example below, the epidemiological week will be calculated based on a starting day of Monday (i.e. Sunday will be 1, Monday will be 2, Tuesday will be 3, etc.)



```
Field OnsetDate
	After
		//add code here
		ASSIGN SurveillanceWeek = EPIWEEK( OnsetDate,2)

	End-After
End-Field
```

Figure 7.63: Check code syntax for EPIWEEK function with beginning week parameter

Proper Check Code Syntax

Using the proper check code syntax is important based on the field type. Here are some examples of the proper syntax to use based on the field type;

- Assign the 'Age' field (numeric field type) the value 24
  - **ASSIGN Age = 24**

- Assign the 'Ill' field (Yes/No field type) the value No
  - **ASSIGN Ill = (-)**

- Assign the 'AteChicken' field (checkbox field type) the value Yes
  - **ASSIGN AteChicken = (+)**

- Assign the 'DateOfInterview' field (Date field type) the value 5/5/2012
  - **ASSIGN DateOfInterview = 5/5/2012**

- Assign the 'CaseStatus' field (legal values field type) the value "Confirmed"
  - **ASSIGN CaseStatus = "Confirmed"**

- Assign the 'Gender' field (Comment Legal field) the value "F". The field is coded as M-Male, F-Female.
  - **ASSIGN Gender = "M"**

THIS PAGE IS

INTENTIONALLY BLANK.

4. Enter: Entering and Editing Data in Epi Info™

Introduction

The Enter module allows users to enter data into forms constructed in Form Designer. Enter conducts the data entry process according to the settings and Check Code specified in Form Designer. Information entered into the form is stored in Microsoft Access or Microsoft SQL server data tables.

Enter can be used to enter new data, modify existing data, or search for records. When data are entered into a form, the data table inside the project is populated. The Find function allows records to be located based on a series of matched variables. As data are entered, the cursor moves according to the tab order from field-to-field, page-to-page, and saves data when navigating to a new page. If you try to exit a page before data is saved, you will be prompted to save the data (Yes or No).

Open a Project and Form

To access the Enter program, click the **Enter Data** button on the Epi Info™ main menu or select **Tools > Enter Data** from the toolbar. In Form Designer, access the Enter module by clicking the **Enter Data** button in the toolbar. This button is a convenient method to switch between designing a form and entering data to test the form design.

There are three options available to open a project and form. The following example uses the **Food History** form available in the **EColi** project.

1. From the toolbar, select **File > Open Form.**

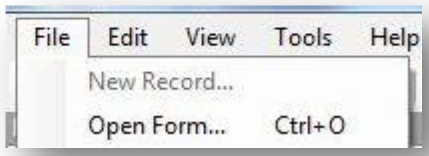


Figure 7.190: Open Form

2. Alternatively, click **Open Form** from the toolbar,

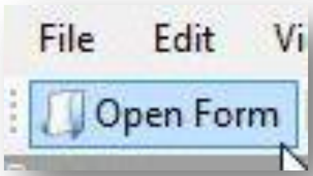
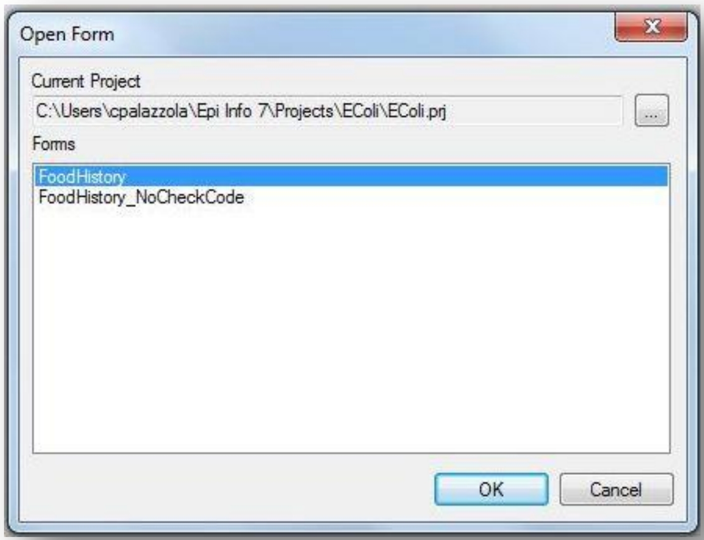


Figure 7.191: Open Form

3. These two options will display the **Open Form** dialog box.



C:\Users\cpalazzola\Epi Info 7\Projects\EColi\EColi.prj

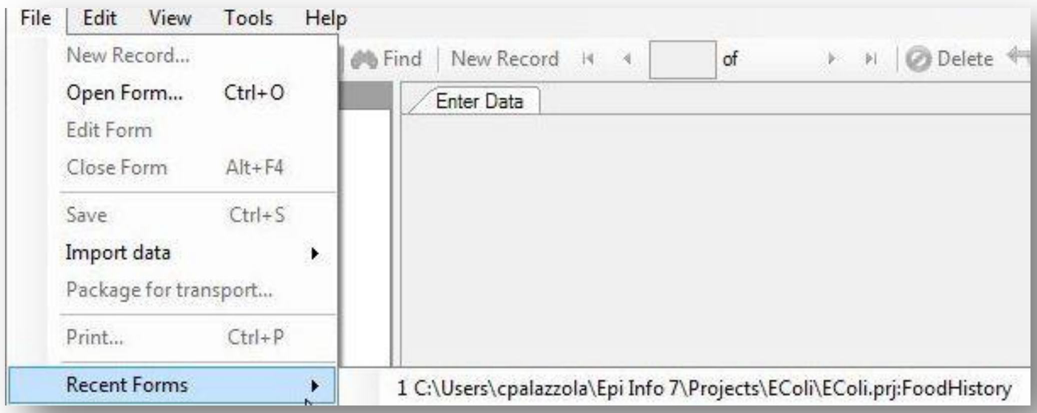
FoodHistory
FoodHistory_NoCheckCode

Figure 7.192: Open Form dialog box

4. Click the **ellipses** button next to the Current Project field to browse for a project. Select **EColi** from the folder. The project file path displays in the Current Project field and a list of available forms in that project appear in the Forms field.
5. Select the desired form (**FoodHistory)** from the Forms field.
6. Click **OK**. The project and form display with the first record in the project appearing on the canvas.

For quick access to a project recently opened with the Enter module, use the Recent Forms option.

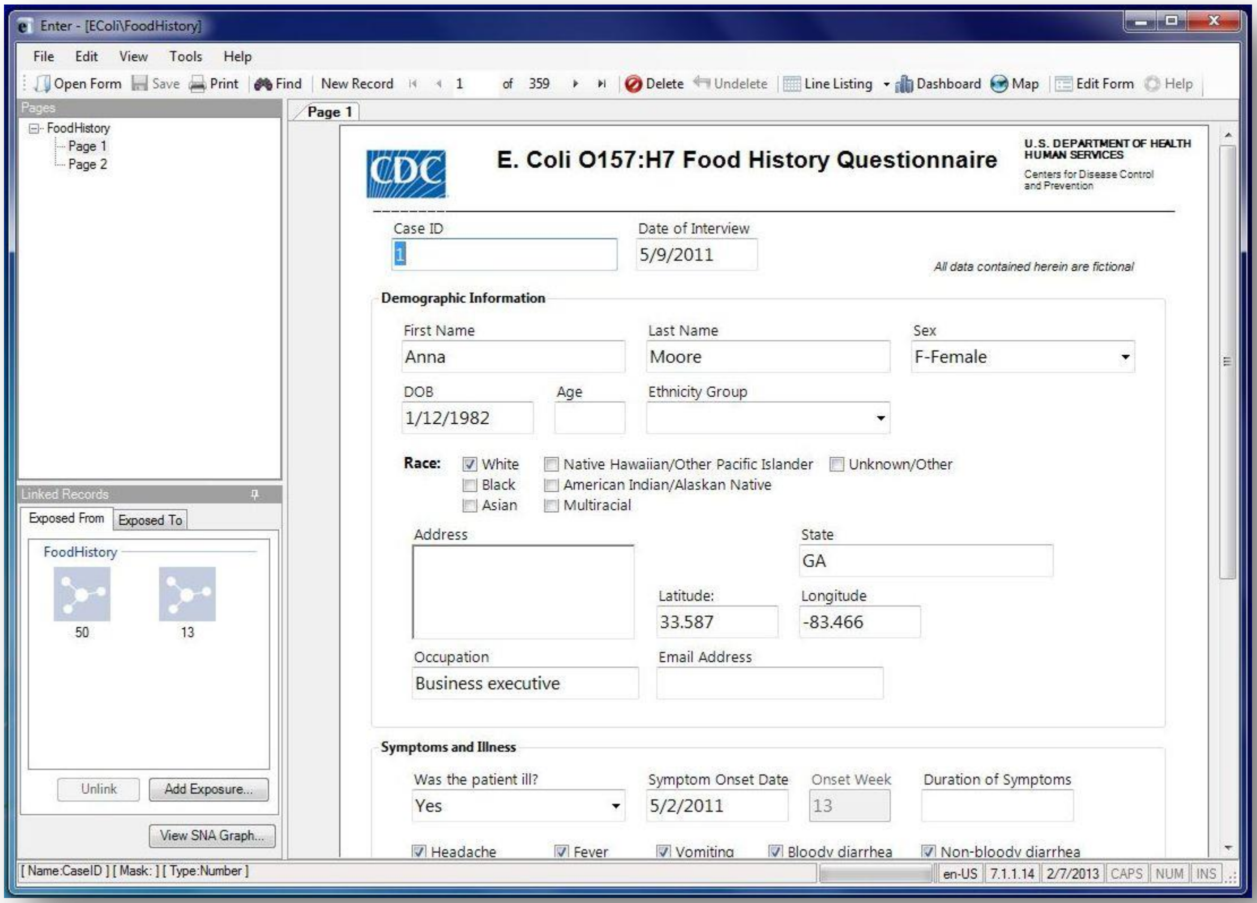
1. From the toolbar, select **File** > **Recent Forms**.



1 C:\Users\cpalazzola\Epi Info 7\Projects\EColi\EColi.prj:FoodHistory

Figure 7.193: Open Recent Form

2. Select the path file containing the desired project and form. The project and form display with the first record in the project appearing on the canvas.



E. Coli O157:H7 Food History Questionnaire

Case ID: 1
Date of Interview: 5/9/2011
All data contained herein are fictional

Demographic Information

First Name: Anna
Last Name: Moore
Sex: F-Female
DOB: 1/12/1982
Age:
Ethnicity Group:
Race: [✓ White] Black Asian Native Hawaiian/Other Pacific Islander American Indian/Alaskan Native Multiracial Unknown/Other
Address:
State: GA
Latitude: 33.587
Longitude: -83.466
Occupation: Business executive
Email Address:

Symptoms and Illness

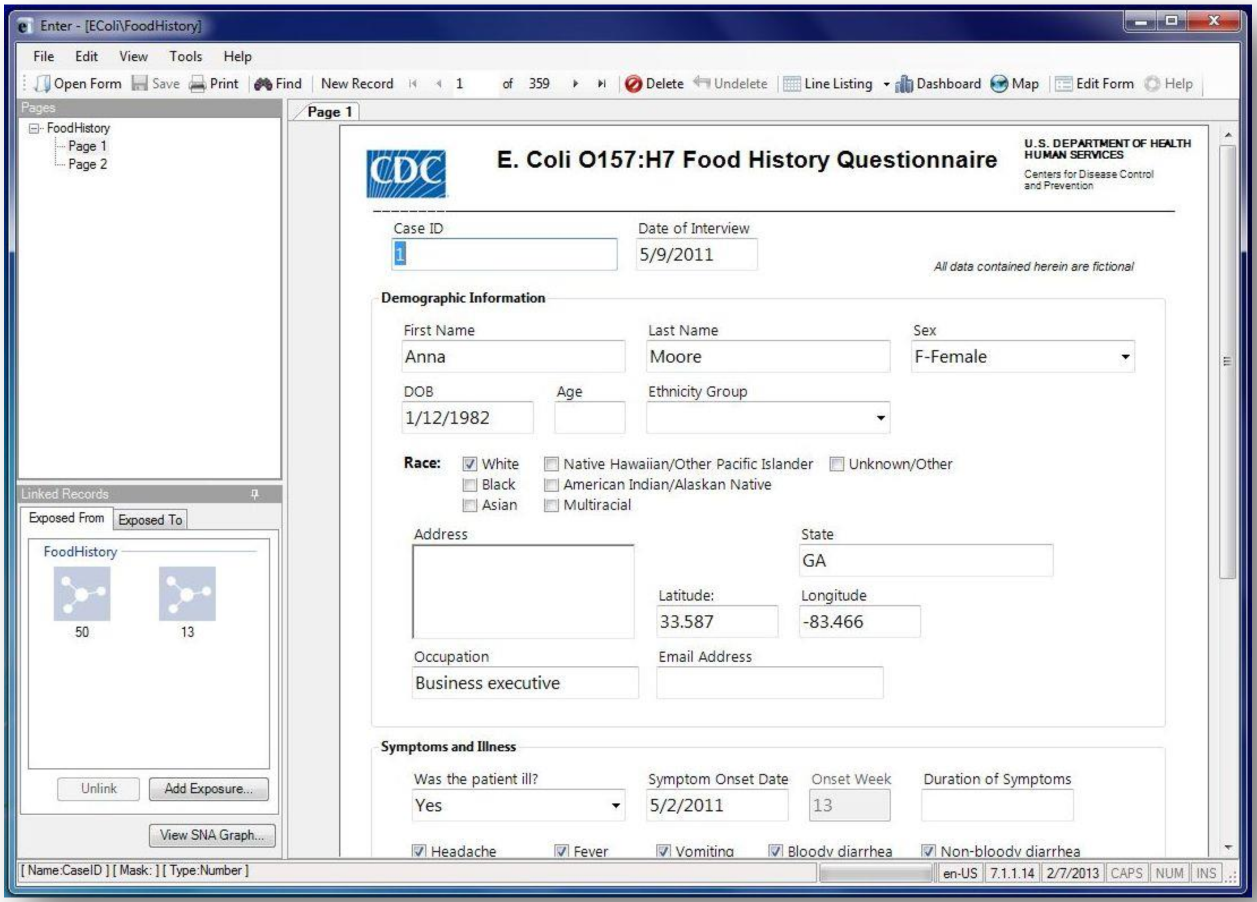
Was the patient ill? Yes
Symptom Onset Date: 5/2/2011
Onset Week: 13
Duration of Symptoms:
[✓ Headache] [✓ Fever] [✓ Vomiting] [✓ Bloody diarrhea] [✓ Non-bloody diarrhea]

[Name:CaseID] [Mask:] [Type:Number]
en-US | 7.1.1.14 | 2/7/2013 | CAPS | NUM | INS

Figure 7.194: Enter Workspace

Navigating the Enter Workspace

There are five main areas in the Enter workspace: Pages, Linked Records, Canvas, Toolbar, and Status Bar. The figure below displays all five areas with information contained in the **FoodHistory** form of the **EColi** project.


Enter - [EColi\FoodHistory]
File Edit View Tools Help
Open Form Save Print Find New Record 1 of 359 Delete Undelete Line Listing Dashboard Map Edit Form Help
Pages
FoodHistory
Page 1
Page 2
Linked Records
Exposed From Exposed To
FoodHistory
50
13
Unlink Add Exposure...
View SNA Graph...
Page 1
CDC
U.S. DEPARTMENT OF HEALTH HUMAN SERVICES
Centers for Disease Control and Prevention

E. Coli O157:H7 Food History Questionnaire

Case ID: 1
Date of Interview: 5/9/2011

All data contained herein are fictional

Demographic Information

First Name: Anna
Last Name: Moore
Sex: F-Female
DOB: 1/12/1982
Age:
Ethnicity Group:

Race: [✓ White] Black Asian Native Hawaiian/Other Pacific Islander American Indian/Alaskan Native Multiracial Unknown/Other

Address:
State: GA
Latitude: 33.587
Longitude: -83.466
Occupation: Business executive
Email Address:

Symptoms and Illness

Was the patient ill? Yes
Symptom Onset Date: 5/2/2011
Onset Week: 13
Duration of Symptoms:

[✓ Headache] [✓ Fever] [✓ Vomiting] [✓ Bloody diarrhea] [✓ Non-bloody diarrhea]

[Name:CaseID] [Mask:] [Type:Number] en-US 7.1.1.14 2/7/2013 CAPS NUM INS

Figure 7.195: Enter Workspace

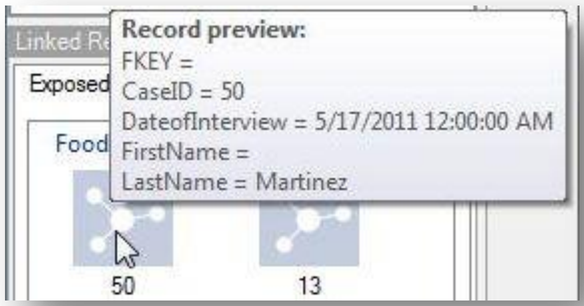
Pages

The **Pages** panel in the top left corner lists all pages associated with the current form. Users may navigate to different pages in the form by selecting the desired page in the **Pages** panel.

Linked Records

The **Linked Records** panel in the bottom left is used to conduct contact tracing. It contains information about which records have been exposed to or from the current record.

1. **Exposed From** – Displays additional records that the current record has been exposed or affected from. The **Exposed From** icons appear with the Case ID number of the record. Place the mouse over the record to preview the record information.



Record preview:
FKEY =
CaseID = 50
DateofInterview = 5/17/2011 12:00:00 AM
FirstName =
LastName = Martinez

Figure 7.196: Exposed From - Record Preview

- Double-click on the record to jump to that record and view additional information

2. **Exposed To** - Displays other records that the current record has been in contact with or possibly affected since acquiring the illness. **The Exposed To** icons appear with the Case ID number of the record.

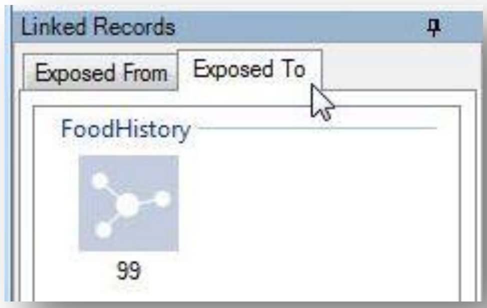


Figure 7.197: Exposed To

- Similar to the **Exposed From** tab, users can preview information or jump to the record by double-clicking the icon.

3. **Add Exposure** – Identifies additional exposures (To and From) for the current record. To add an **Exposure To**, click **Add Exposure** in the **Exposure To** tab. To add an **Exposure From**, click **Add Exposure** in the **Exposure From** tab.

   - Select the form that the exposure record is contained. Click **OK**.

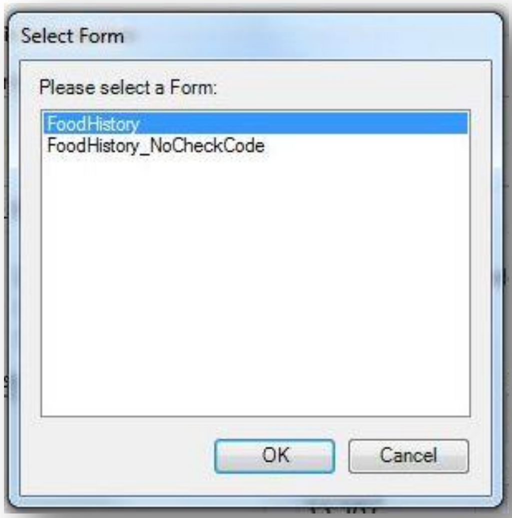


Figure 7.198: Add Exposure from Form

- Find the desired exposure information with Find (information regarding the Find function is located in the Find Records section). Double click on search results to add the exposure to the record.

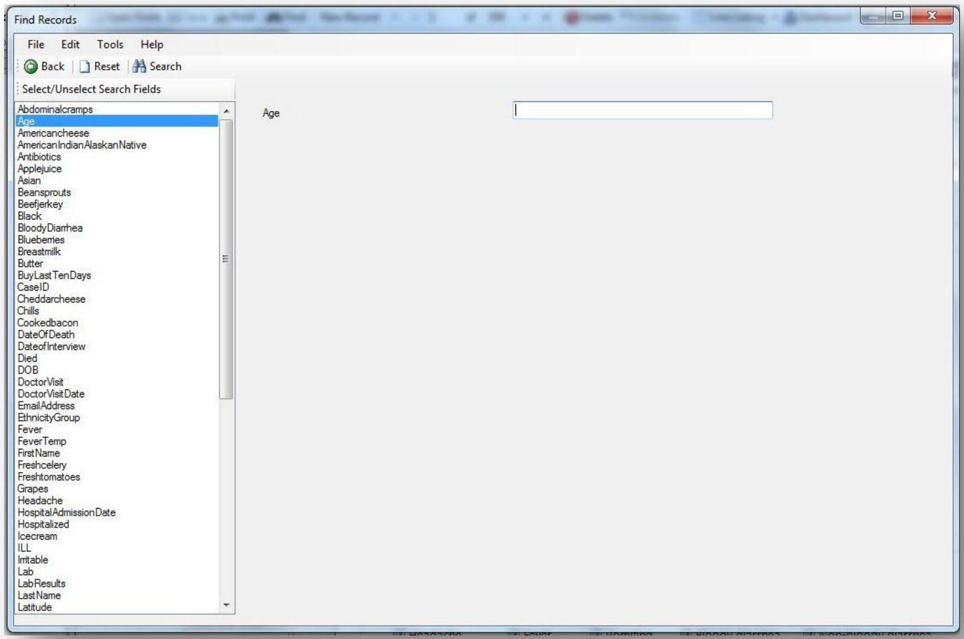


Figure 7.199: Add Exposure from Record

4. **SNA graph** – Social Network Analysis (SNA) displays the exposed to and from records graphically with the current record in red.

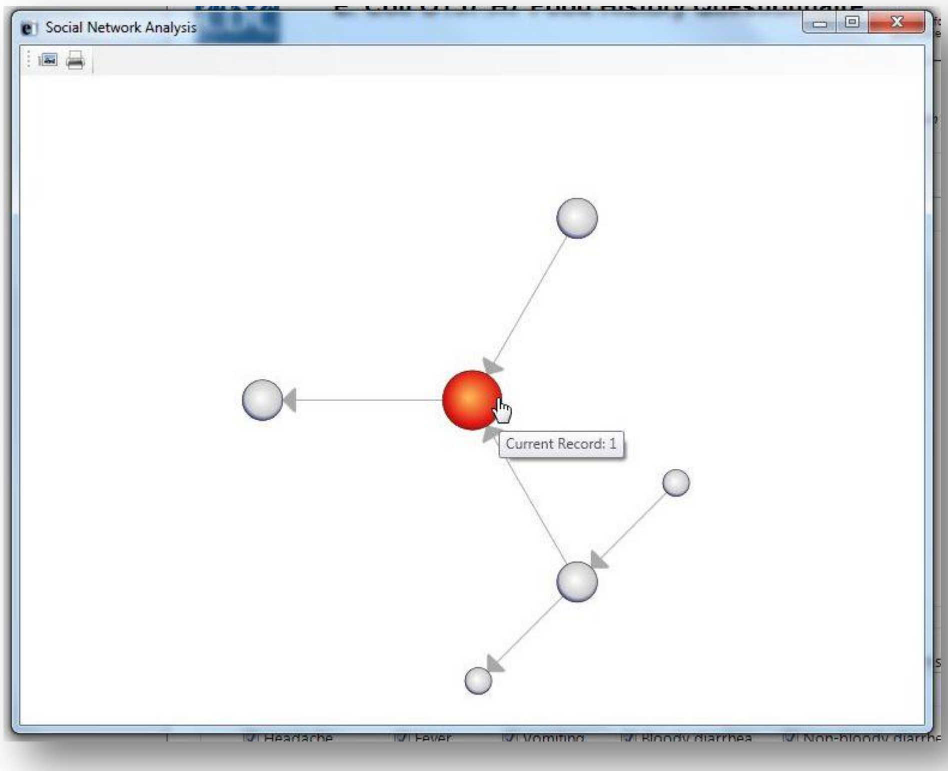


Figure 7.200: SNA Graph

- The SNA graph directionally shows the flow of exposure at multiple levels. For example, the current record has two records that it has been exposed from; however, one of those records has been exposed from two more records.
- The SNA graph also displays that the current record has been exposed to another record. This is consistent with the information contained in the **Exposed From** and **Exposed To** tabs.
- The SNA graph can be saved as an image or printed using the buttons in the top left corner of the screen.

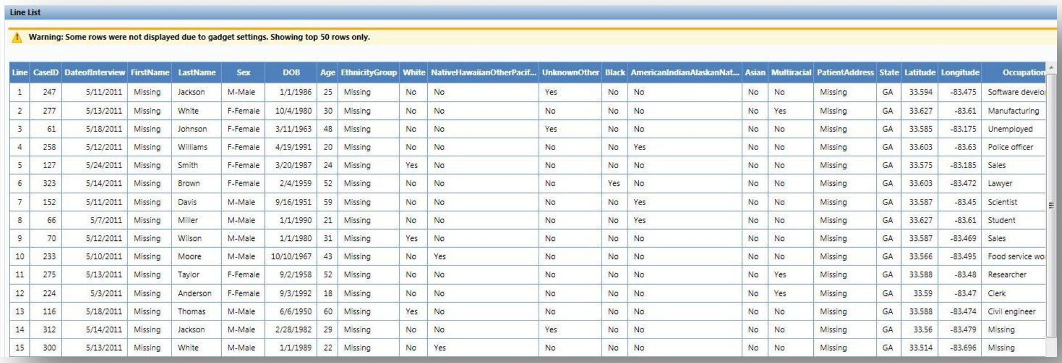
Canvas

The **Canvas** shows the current page in the form with all record information displayed. The canvas is where data entry and editing occurs.

Toolbar

The **Toolbar** at the top of the screen contains buttons to navigate through the records, create new records, open the Dashboard, open the mapping tool, save the current record, view records in a line list, and print.

- **Line List** – allows users to view and search records using the Dashboard interface. Also, users can view the data in MS Excel and HTML formats. For additional information, reference the Visual Dashboard section of the user guide.



Line List

Warning: Some rows were not displayed due to gadget settings. Showing top 50 rows only.

Line	CaseID	DateofInterview	FirstName	LastName	Sex	DOB	Age	EthnicityGroup	White	NativeHawaiianOtherPacif...	UnknownOther	Black	AmericanIndianAlaskanNat...	Asian	Multiracial	PatientAddress	State	Latitude	Longitude	Occupation
1	247	5/11/2011	Missing	Jackson	M-Male	1/1/1986	25	Missing	No	No	Yes	No	No	No	No	Missing	GA	33.594	-83.475	Software develo
2	277	5/13/2011	Missing	White	F-Female	10/4/1980	30	Missing	No	No	No	No	No	No	Yes	Missing	GA	33.627	-83.61	Manufacturing
3	61	5/18/2011	Missing	Johnson	F-Female	3/11/1963	48	Missing	No	No	Yes	No	No	No	No	Missing	GA	33.585	-83.175	Unemployed
4	258	5/12/2011	Missing	Williams	F-Female	4/19/1991	20	Missing	No	No	No	No	Yes	No	No	Missing	GA	33.603	-83.63	Police officer
5	127	5/24/2011	Missing	Smith	F-Female	3/20/1987	24	Missing	Yes	No	No	No	No	No	No	Missing	GA	33.575	-83.185	Sales
6	323	5/14/2011	Missing	Brown	F-Female	2/4/1959	52	Missing	No	No	No	Yes	No	No	No	Missing	GA	33.603	-83.472	Lawyer
7	152	5/11/2011	Missing	Davis	M-Male	9/16/1951	59	Missing	No	No	No	No	Yes	No	No	Missing	GA	33.587	-83.45	Scientist
8	66	5/7/2011	Missing	Miller	M-Male	1/1/1990	21	Missing	No	No	No	No	Yes	No	No	Missing	GA	33.627	-83.61	Student
9	70	5/12/2011	Missing	Wilson	M-Male	1/1/1980	31	Missing	Yes	No	No	No	No	No	No	Missing	GA	33.587	-83.469	Sales
10	233	5/10/2011	Missing	Moore	M-Male	10/10/1967	43	Missing	No	Yes	No	No	No	No	No	Missing	GA	33.566	-83.495	Food service wo
11	275	5/13/2011	Missing	Taylor	F-Female	9/2/1958	52	Missing	No	No	No	No	No	No	Yes	Missing	GA	33.588	-83.48	Researcher
12	224	5/3/2011	Missing	Anderson	F-Female	9/3/1992	18	Missing	No	No	No	No	No	No	Yes	Missing	GA	33.59	-83.47	Clerk
13	116	5/18/2011	Missing	Thomas	M-Male	6/6/1950	60	Missing	Yes	No	No	No	No	No	No	Missing	GA	33.588	-83.474	Civil engineer
14	312	5/14/2011	Missing	Jackson	M-Male	2/28/1982	29	Missing	No	No	Yes	No	No	No	No	Missing	GA	33.56	-83.479	Missing
15	300	5/13/2011	Missing	White	M-Male	1/1/1989	22	Missing	No	Yes	No	No	No	No	No	Missing	GA	33.514	-83.696	Missing

Figure 7.201: Line List View

- **Print** – allows the user to print the form and specify customized parameters.

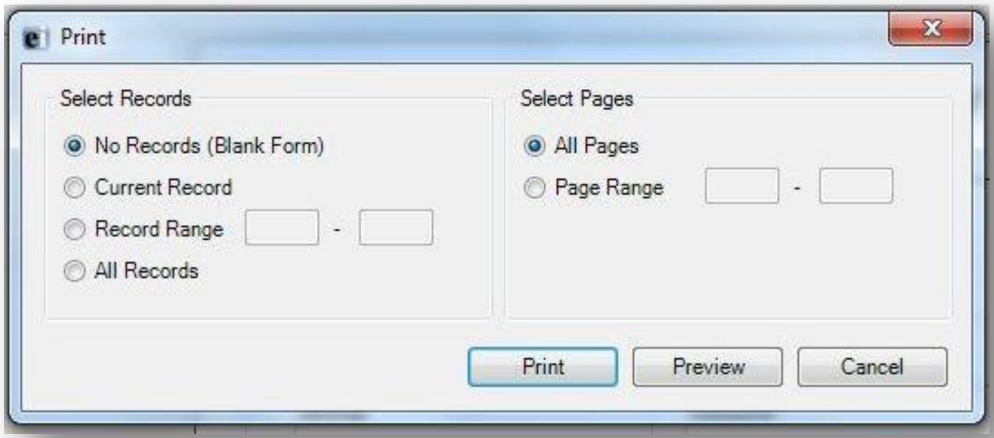
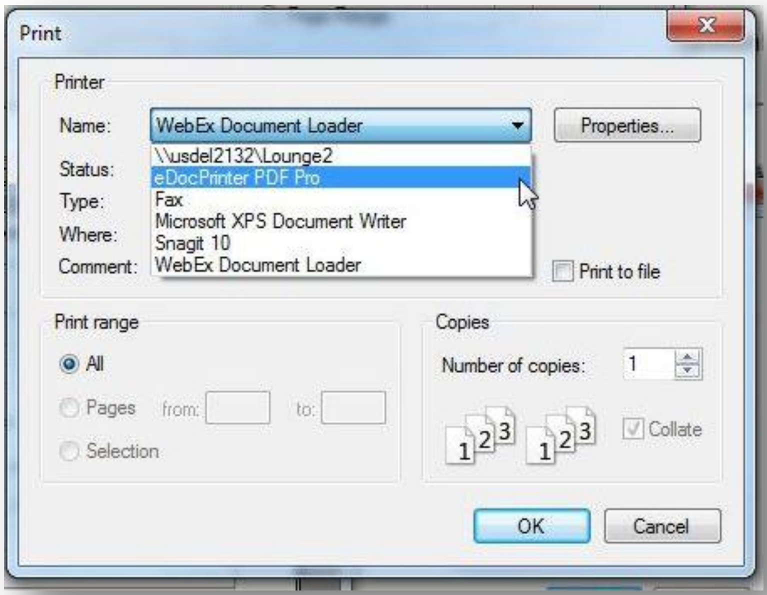


Figure 7.202: Print dialog box

- To create a PDF version of the form using the print function select **Preview**. In the **Print Preview** window select **Print** and chose the appropriate PDF writer as the printer.

Note: The ability to print to a PDF requires PDF writer software.


Print

Printer

Name: WebEx Document Loader

Status:

Type:

Where:

Comment:

\\usdel2132\Lounge2
eDocPrinter PDF Pro
Fax
Microsoft XPS Document Writer
Snagit 10
WebEx Document Loader

Print to file

Print range: All [✓], Pages from: to:, Selection

Copies: Number of copies: 1, Collate [✓]

Figure 7.203: Create PDF

- **Navigation Buttons** - Navigation is provided for using the New Record button and navigation buttons for, next (▶), previous (◀), first (⏮), and last (⏭) records. The user may also manually type the record ID number into the record list to jump to that record. When entering data into a child form, use the "Back" button to return to the parent form.

- **Maps** – Entering Maps from the Enter module allows for some additional functionality between Maps and the project data. The additional functionality allows the user to view case data in the Enter module from the Map window. The following example demonstrates how to create a case cluster map using the FoodHistory form data in the Ecoli project. For additional information regarding maps, reference the Maps section of the user guide.

1. Click **Maps** from the toolbar. The Maps main window appears.

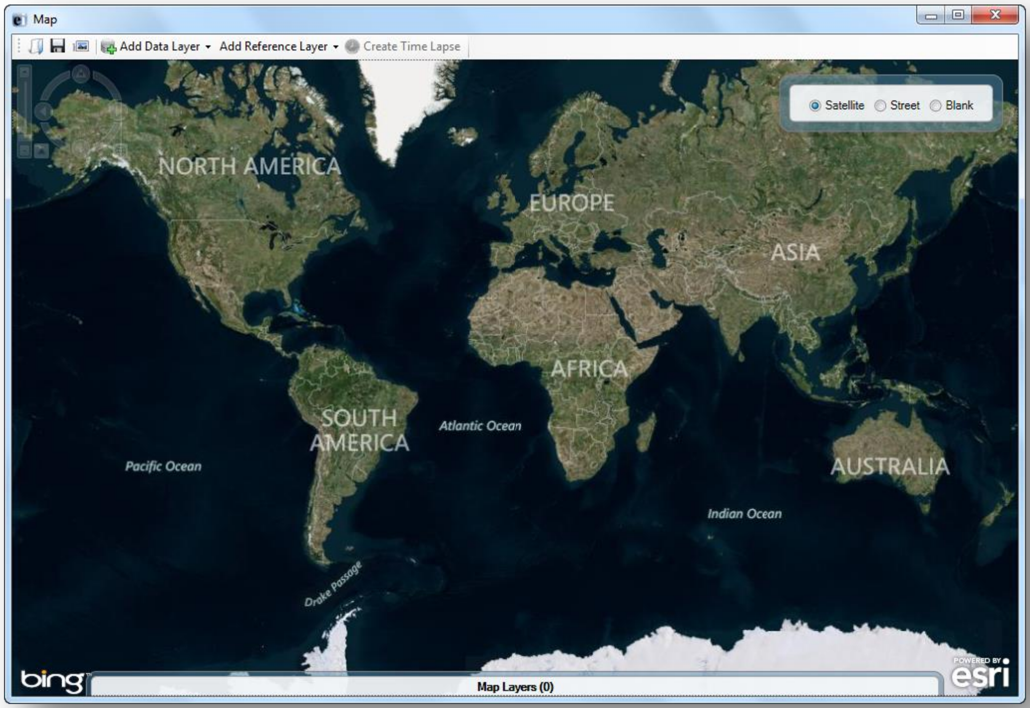


Figure 7.204: Main Map Window

2. Select **Add Data Layer > Case Cluster** from the toolbar.

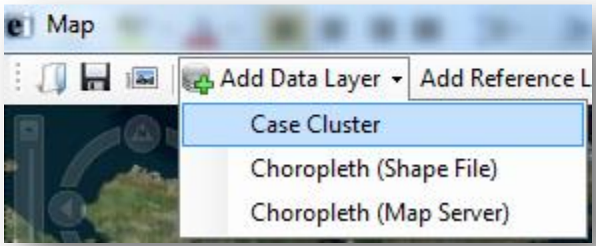


Figure 7.205: Case Cluster

3. In the external data dialog box, select **No**.

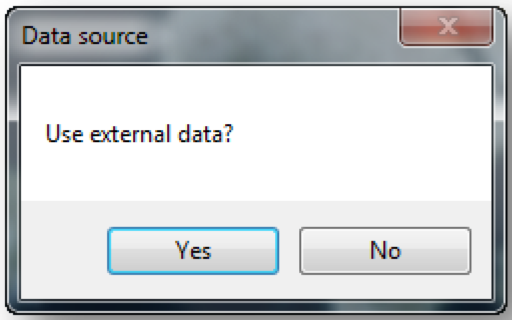


Figure 7.206: Data Source

4. Select the field containing the Latitude coordinates (**Latitude)** from the Latitude drop-down list.
5. Select the field containing the Longitude coordinates (**Longitude)** from the Longitude drop-down list.

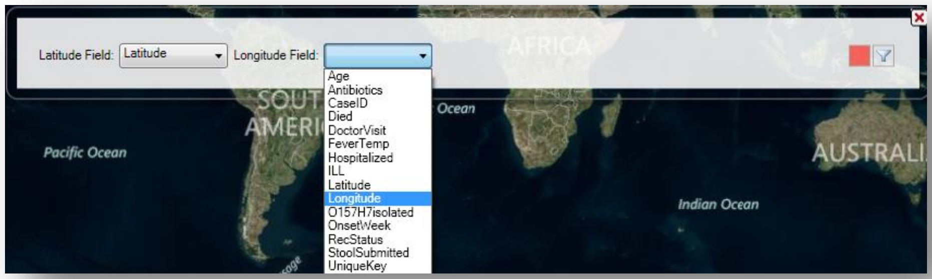


Figure 7.207: Latitude and Longitude Fields

6. The case cluster map appears displaying each record's location on the map.

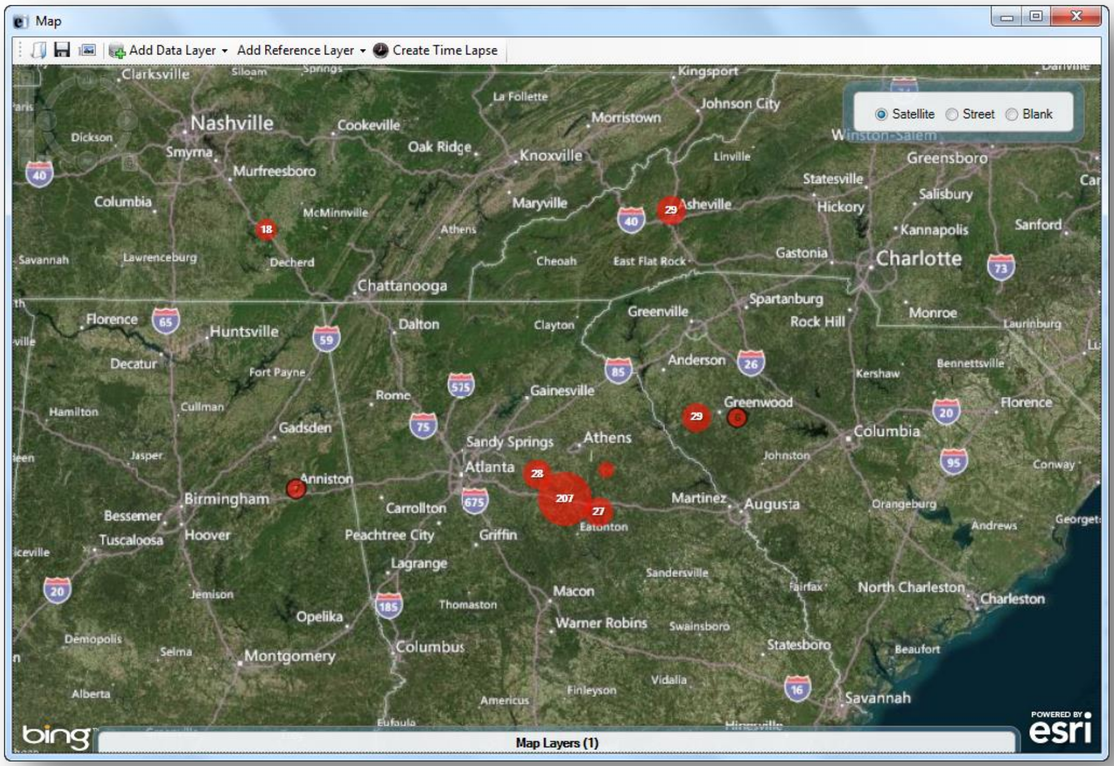
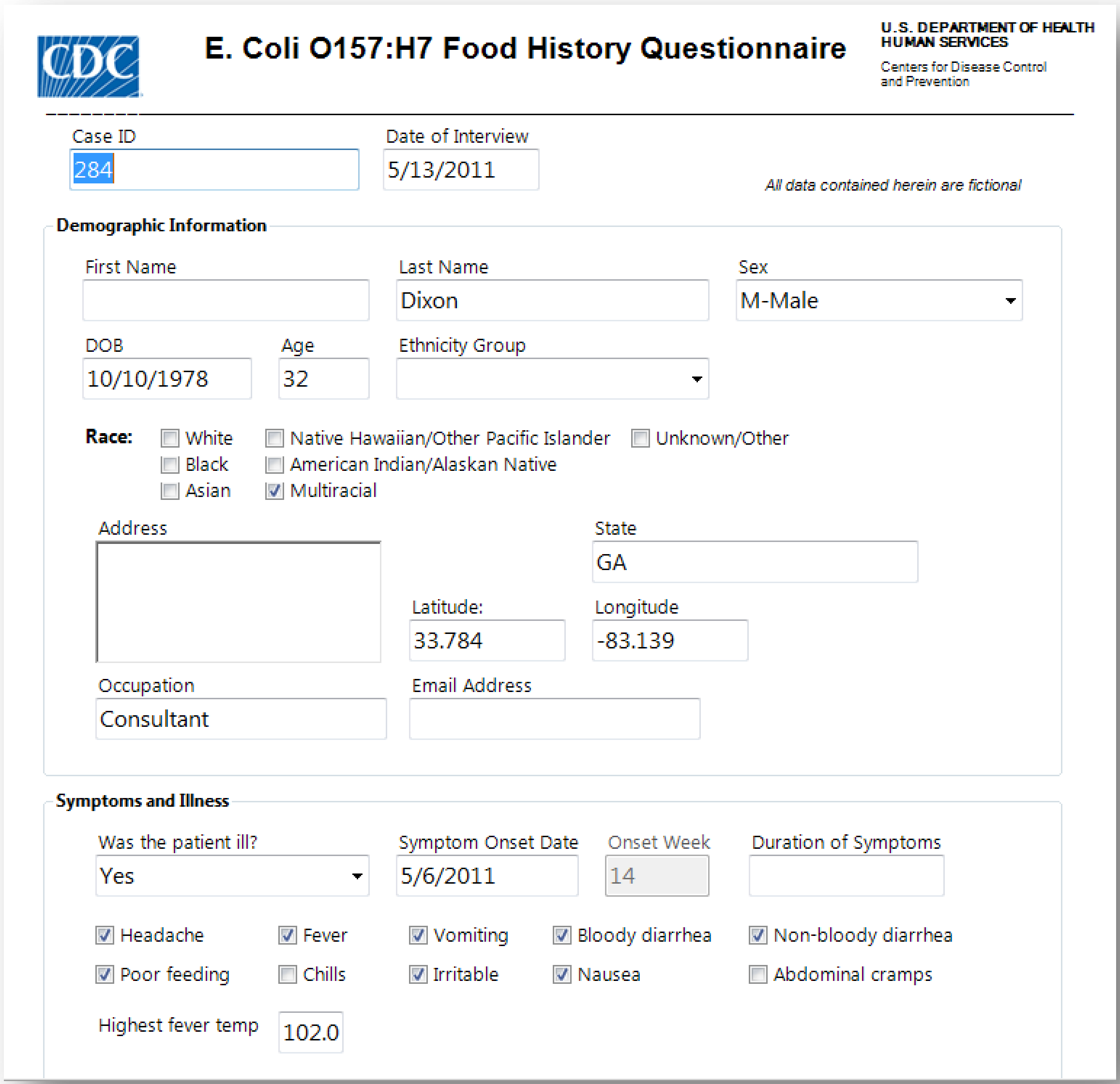


Figure 7.208: Case Cluster Map

7. To view the information associated with each record on the map, double-click on a single red dot. The record appears in the Enter data window.

Note: Zoom in on the record of interest to separate the case clusters into single dots.



E. Coli O157:H7 Food History Questionnaire

U.S. DEPARTMENT OF HEALTH HUMAN SERVICES
Centers for Disease Control and Prevention

Case ID: 284
Date of Interview: 5/13/2011

All data contained herein are fictional

Demographic Information

First Name:
Last Name: Dixon
Sex: M-Male
DOB: 10/10/1978
Age: 32
Ethnicity Group:

Race: ☐ White ☐ Native Hawaiian/Other Pacific Islander ☐ Unknown/Other ☐ Black ☐ American Indian/Alaskan Native ☐ Asian ☑ Multiracial

Address:
State: GA
Latitude: 33.784
Longitude: -83.139
Occupation: Consultant
Email Address:

Symptoms and Illness

Was the patient ill? Yes
Symptom Onset Date: 5/6/2011
Onset Week: 14
Duration of Symptoms:

☑ Headache ☑ Fever ☑ Vomiting ☑ Bloody diarrhea ☑ Non-bloody diarrhea
☑ Poor feeding ☐ Chills ☑ Irritable ☑ Nausea ☐ Abdominal cramps

Highest fever temp: 102.0

Figure 7.209: Case Record

Status Bar

The **Status Bar** at the bottom of the screen displays selected information about the field that the cursor is currently located. The status bar can be hidden by deselecting **View > Status Bar** in the toolbar.

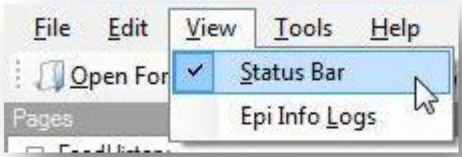


Figure 7.210: Status Bar

Check Code Options

The Enter module allows the user to modify the Check Code features when entering data. Both Check Code options are available under Tools in the toolbar and are enabled by default.

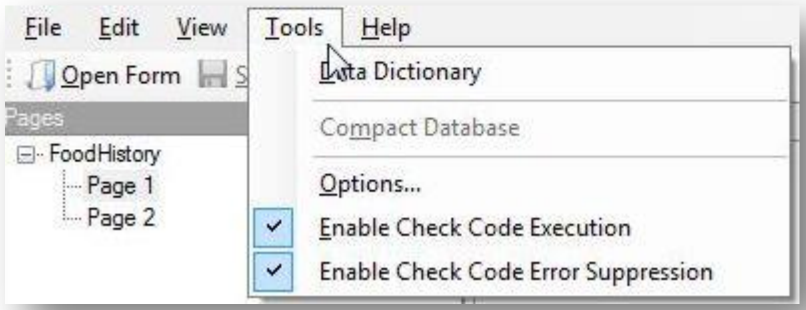


Figure 7.211: Check Code Options

- **Enable Check Code Executions** – Unchecking this option will disable all check code when entering data into the form
- **Enable Check Code Error Suppression** - Unchecking this option will not display error messages in check code

Import Data

There are multiple options available to import data from external sources into the Enter module. These options are available under **File > Import Data** in the toolbar.

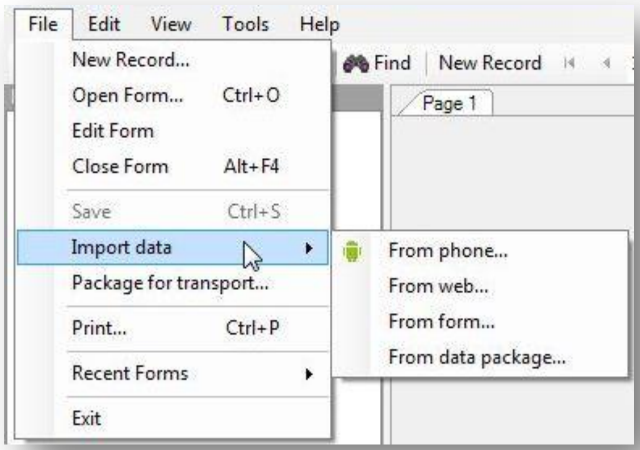
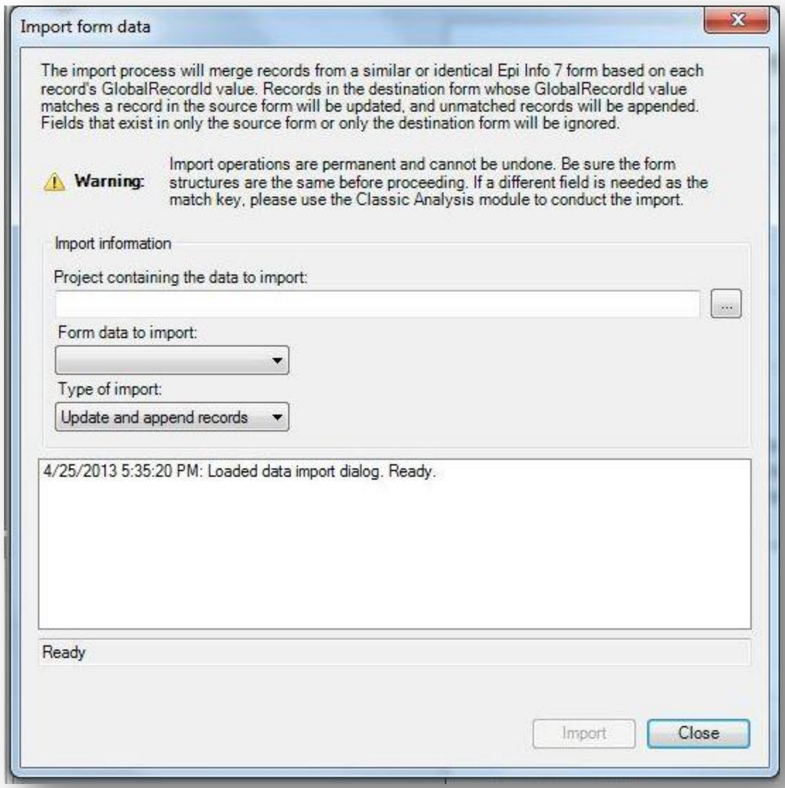


Figure 7.212: Import Data

- **From form** – This feature requires a Epi Info project file (.prj) and data (.mdb if data tables formed in Microsoft Access).

- Browse for the Epi Info project file by clicking the ellipses next to the **Project containing the data to import** field. Select the .prj file
- Select the **Form** from the **Form data to import** drop-down list.


Import form data

The import process will merge records from a similar or identical Epi Info 7 form based on each record's GlobalRecordId value. Records in the destination form whose GlobalRecordId value matches a record in the source form will be updated, and unmatched records will be appended. Fields that exist in only the source form or only the destination form will be ignored.

Warning: Import operations are permanent and cannot be undone. Be sure the form structures are the same before proceeding. If a different field is needed as the match key, please use the Classic Analysis module to conduct the import.

Import information

Project containing the data to import:

Form data to import:

Type of import: Update and append records

4/25/2013 5:35:20 PM: Loaded data import dialog. Ready.

Ready

Figure 7.213: Import Data from Form

- Select the **Import Type** from the **Type of import** drop-down list.

  - **Update and append records** – Records in the destination form containing the same GlobalRecordId values as the source form will be updated (overwritten) if there is new information. However, values in the destination table will never be overwritten with a null or missing value. All other records will be appended (added) to the end of the table.

  - **Update records only** – Records in the destination form containing the same GlobalRecordId values as the source form will be updated (overwritten) if there is new information. However, values in the destination table will never be overwritten with a null or missing value. All other records will be ignored.

  - **Append records only** – All records in the source form will be appended (added) to the end of the table. No records will be updated (overwritten).

- Click **Import**. *Import Complete* will appear at the bottom of the dialog box upon successful completion of the import.

- Click **Close**. The data is imported into the Enter module.

- **From mobile device** – for information regarding this feature, reference the Companion for Android section of this user guide.
- **From web** – for information regarding this feature, reference the Web Survey section of this user guide.
- **From data package** – for information regarding this feature, reference the Data Packager section of this user guide.

Enter Data in a Form

New Record

1. With a form and project open, select **New Record** from the toolbar or **File > New Record**. A blank record is added to the end of the record list and displays in the canvas. Place the **cursor** in the data entry field.

2. Enter **data** into each field. Use the **Tab** or **Enter** key to move to the next field. Moving from one field to another will execute Check Code.

   - As records are navigated, data are saved automatically. After completing the last field on a page, the cursor automatically jumps to the next page in the sequence.
   - Field properties that were defined in Form Designer, (e.g., Read Only, Required Range) are enforced during data entry.
   - Drop-down lists and code tables created in Form Designer are enforced.
   - Multiline Fields automatically scroll when filled with text to hold up to 1 gigabyte of information.
   - Plain text fields automatically scroll when filled with text to hold up to 255 characters per field.

Edit Record

1. Navigate to the desired record using the navigation tools in the toolbar.
2. Place the **cursor** in the desired data entry field.
3. Enter **data** into each field that needs to be updated. Use the **Tab** or **Enter** key to move to the next field. Moving from one field to another will execute Check Code.

Delete Record

1. Navigate to the desired record using the navigation tools in the toolbar.
2. Select the **Delete** button in the toolbar, or select **Edit > Mark as Deleted**.
3. The record is grayed out and removed from any analysis performed in Visual Dashboard and Classical Analysis.

Undelete Record

1. Navigate to the desired deleted record using the navigation tools in the toolbar.
2. Select the **Undelete** button in the toolbar, or select **Edit > Undelete**.
3. The record is enabled and will be available for analysis in Visual Dashboard and Classical Analysis.

Insert an Image in a Record

1. Click the **Image Field**. The **Select the Picture File** dialog box opens.
2. Select an **image file**.
3. Click **Open**. The selected image appears in the field.

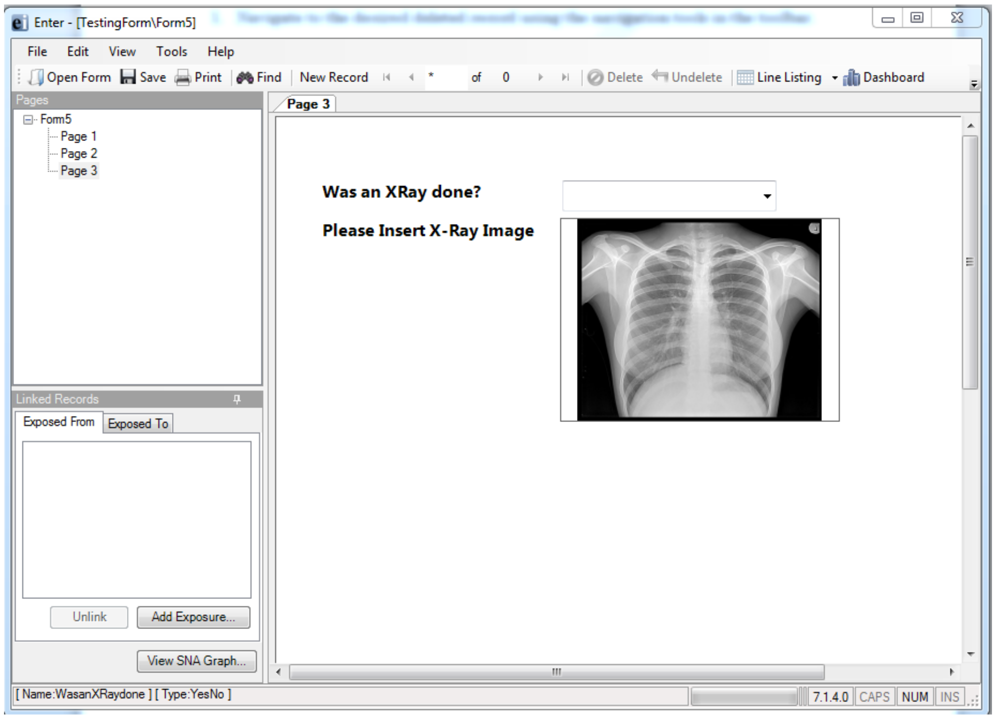


Figure 7.214: Insert Image

Save a Page or Record

The Enter module saves data automatically as the user moves from page-to-page. Data are also saved when navigating to another record. You can move out of a record by tabbing out of the last field of the last page or clicking **New Record** from the toolbar to open a new record. Records can also be saved manually.

There are three ways to save the current record manually:

1. Click **Save** from the toolbar.

2. Select **File > Save**.
3. Press **Ctrl+S** from the keyboard.

Find Records

Find can search for a record matching any field value or combination of field values. If a form has more than 255 fields and more than one associated data table, only the first data table will be searched with the find feature. As a result, only the fields in the first data table will populate in the search results.

Find Records Matching a Specified Criteria

The example below demonstrates how to find records in the **FoodHistory** from in the **EColi** project.

1. From the toolbar, select the **Find** icon or **Edit > Find**. The Find Records window opens.

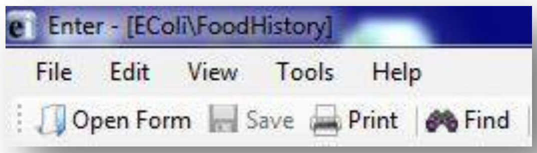


Figure 7.215: Find Record

2. The **Select/Unselect Search Fields** list displays on the left side of the Finds Records window with all fields contained in the form. Click on a field name to select one or more fields. The selected search fields will turn blue and populate the gray area to the right of the list.
3. To unselect a field, click on the selected field highlighted in blue. The field will no longer be highlighted and is removed from the search criteria.

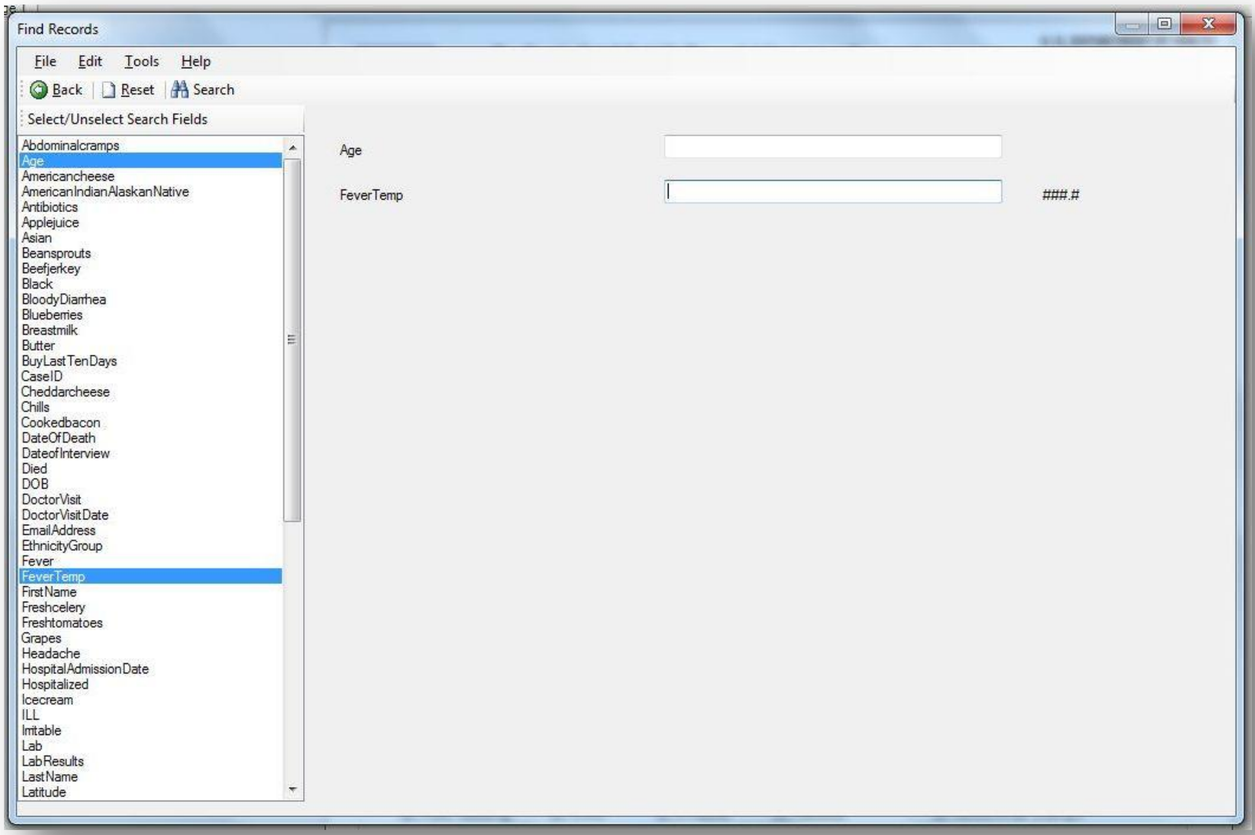


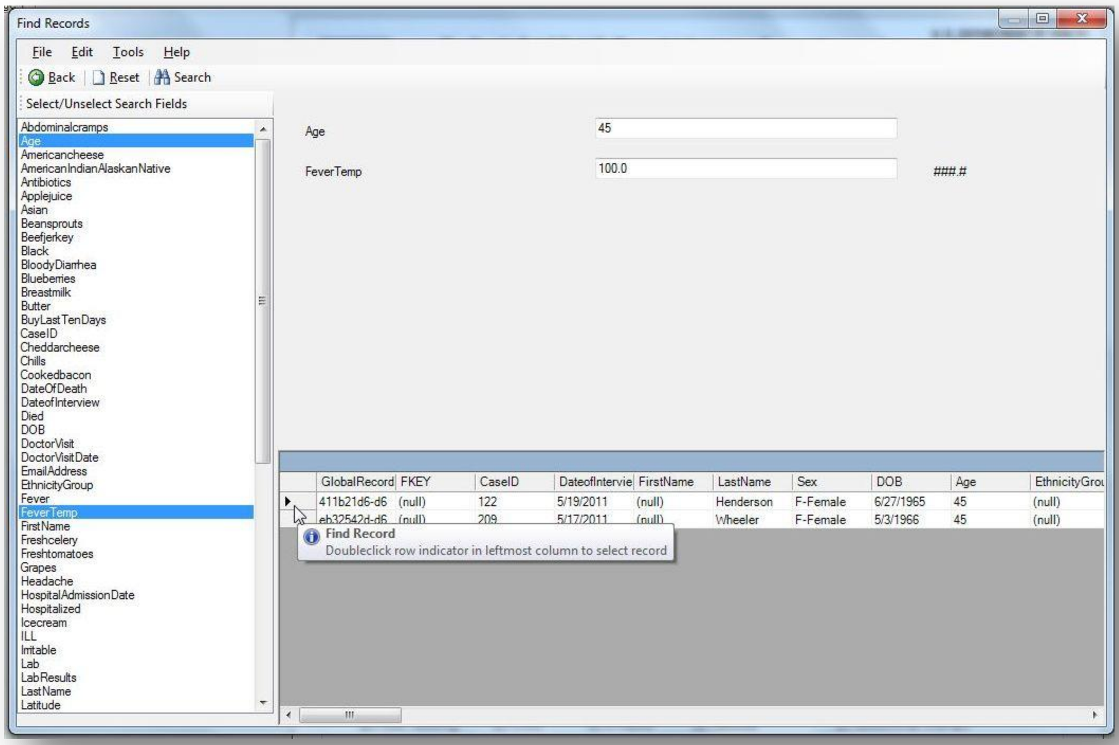
Figure 7.216: Find Records Window

4. Enter the search criteria for each field and click **Search**. Note that certain fields require specific formatting for the search criteria. If more than one field is selected, the search will display records that match all search criteria. This is considered an "AND" search.

   - Yes/No fields can only be searched with the parameters (+), (-) or (.). The parentheses are required.
   - Checkbox can only be searched with the parameters (+) or (-). The parentheses are required.
   - Date fields can only be searched with the date format MM/DD/YYYY.
   - Comment Legal can only be searched with the code (i.e. F- Female, search for F)
   - Option boxes do not store the option labels and therefore must be searched according to the integer that corresponds to each option. In an option box, the first option available corresponds to 0, the second option corresponds to 1, etc. Enter the corresponding value into the search criteria.

- Number fields will only accept values but do not require a decimal point if the field format specifies a decimal value.
- Click **Reset** to begin a new search. Note that more than one field can be searched at a time.
- Click **Back** or the close (X) button to return to the Enter workspace.

5. Open a record by double-clicking the **arrow** on the chosen row. The record displays in the Enter Data workspace.



Age 45

FeverTemp 100.0

	GlobalRecord	FKEY	CaseID	DateofIntervie	FirstName	LastName	Sex	DOB	Age	EthnicityGrou
▶	411b21d6-d6	(null)	122	5/19/2011	(null)	Henderson	F-Female	6/27/1965	45	(null)
	eb32542d-d6	(null)	209	5/17/2011	(null)	Wheeler	F-Female	5/3/1966	45	(null)

Find Record
Doubleclick row indicator in leftmost column to select record

Figure 7.217: Find Records Results

Additional Search Features

1. Embedded text items in multiline and text fields can be found by searching for *word* where "word" is the text string being sought. This type of search is called a Wild Card search. In a Wild Card search, the asterisk represents any letter or string of letters (i.e., a search for DIAGNOSES *heart* would identify all records for a large text field called DIAGNOSES which contained the word "heart)."
2. Dates and numeric fields can be searched using less than or greater than values (<>).

3. Only "OR" can be used to search for matches based on more than one criteria in the same field (i.e. searching last names, Smith OR Doe will return records containing both Smith or Doe. The search field must contain "OR" to perform the desired search. Search criteria cannot be separated by a comma.

4. Searches are not case sensitive and are designed to be more inclusive than exclusive.

Printing Forms

Users can print a blank copy of the form or a copy of the form with data if needed. In order to print the form, please complete the following steps:

1. Open your Epi Info 7 project and form in the Enter Data module.
2. Click on the PRINT option. The following window will be displayed:

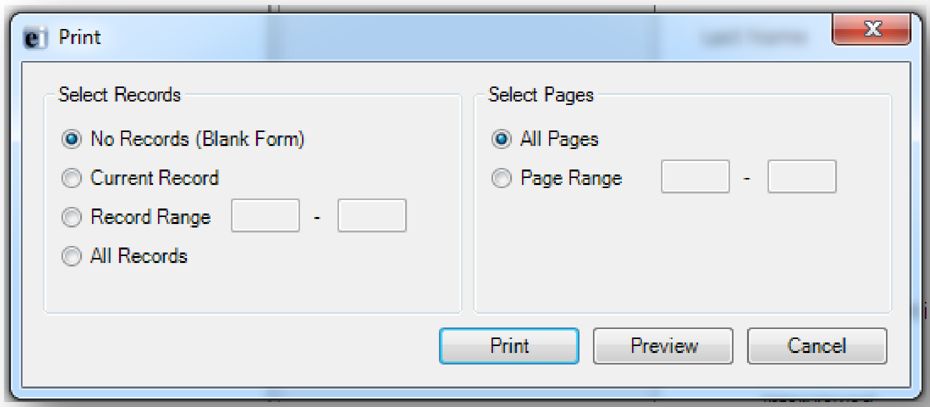


Figure 7.29: Print dialog window

a. To print a copy of the form without including the data that has been entered, just click the PRINT button
b. To print a copy of the form including the data that has been entered for the current record, click on the CURRENT RECORD radio button, and the click the PRINT button. This will print the form with the data for the current record.
c. To print a copy of the form including the data that has been entered for a range of records, click on the RECORD RANGE radio button, and then specify the record number for the records that you want to print the data. When done, click the PRINT button. This will print the form with the data for the records specified.

d. To print a copy of the form including the data for all records, click on the ALL RECORDS radio button, and the click the PRINT button. This will print the form with the data for all records in the database.

3. To print an electronic copy of the form without any data, click on the PREVIEW button.

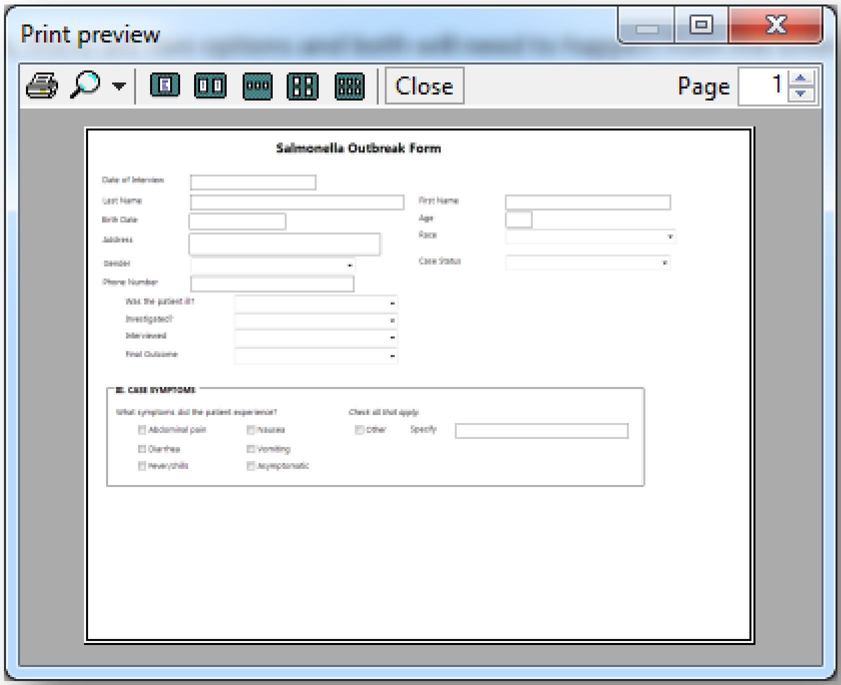


Figure 7.30: Print Preview dialog window

4. Click on the printer icon button.

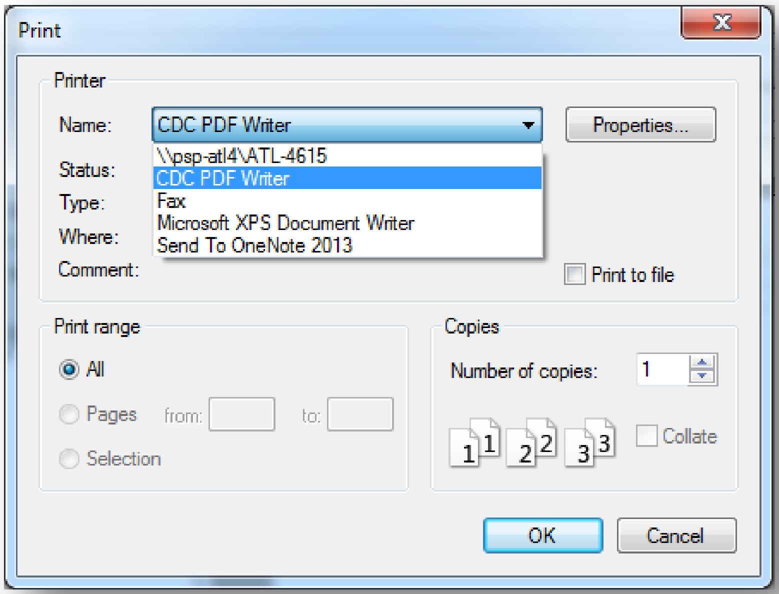


Figure 7.31: Select Printer dialog window

5. Select PDF writer or any other option that allows the creation of an electronic copy of a document.

THIS PAGE IS

INTENTIONALLY BLANK.

5. Web Survey: Surveys through the Internet

Introduction

The Epi Info™ Web Survey System allows the survey designer to collect information from participants over the Internet. The ability to distribute and collect surveys remotely is unique to Epi Info™ 7 and provides survey designers access to a wide variety and number of participants. Survey forms can be published to any properly configured web server hosted by your institution or an outside party. When published, Epi Info™ creates a survey specific Universal Resource Locator (URL) or website address. The survey designer can distribute the URL over email, by posting it on a web page, or using other methods. Participants access the web survey and submit their responses through a web browser or mobile device. After the participant submits the response, the survey designer downloads the response directly into the original Epi Info™ 7 project for analysis.

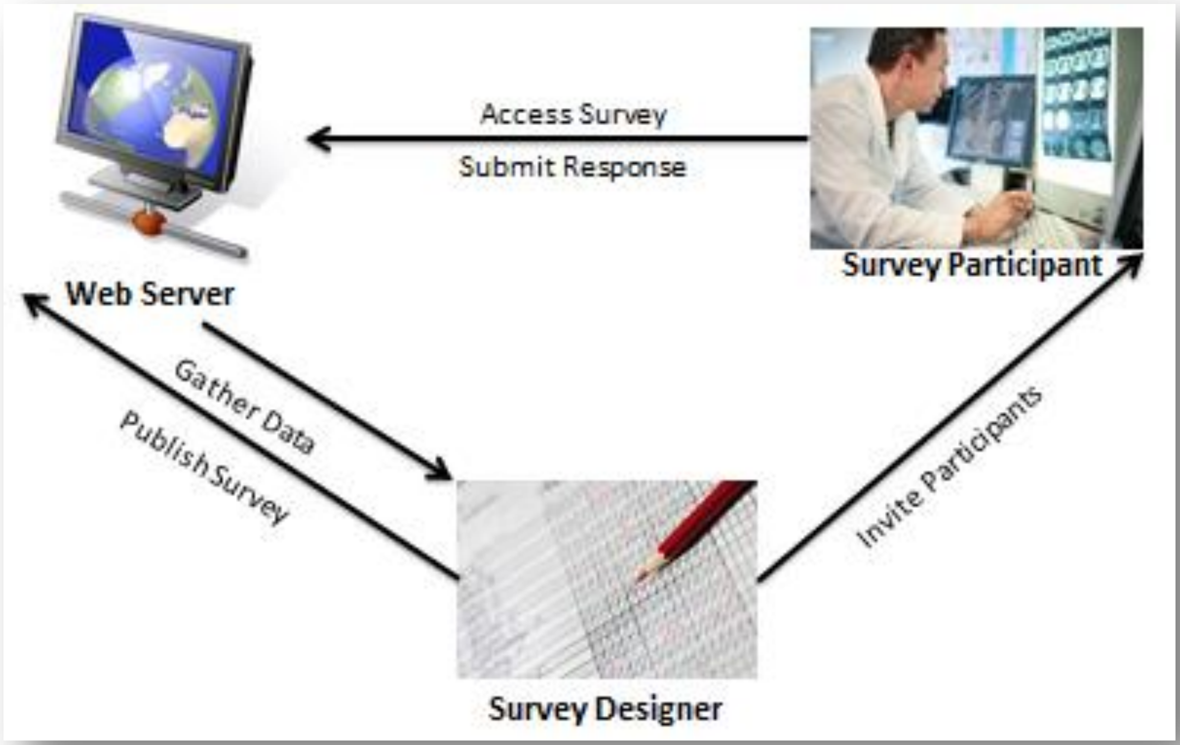


Figure 7.218: Web Survey System

Epi Info™ Web Survey System Deployment

The Epi Info™ Web Survey System is a companion to the Epi Info™ 7 suite of tools, and a separate package apart from Epi Info™ 7. The Web Survey System requires a server running Microsoft Internet Information Services or IIS and a SQL Server database. This database manages the users authorized to publish surveys, published web surveys, and the data submitted by respondents. Without access to the web and database servers that host the Epi Info™ Web Survey System, users will not be able to publish surveys and collect data over the Internet using Epi Info™.

Additional information about the Epi Info™ Web Survey System is located on the Epi Info™ Web Survey page (www.cdc.gov/epiinfo/eiws/index.htm). This site contains setup instructions and links to download the free deployment package kit.

This user guide outlines the steps required to design and publish a form using the Epi Info™ 7 Form Designer tool, and import data submitted by survey participants with the Epi Info™ 7 Enter tool. The technical aspects of configuring the web and database servers are beyond the scope of this guide.

To obtain the necessary keys and settings to use the Web Survey System, contact your Web Survey System administrator, or the person/group that manages the Epi Info™ Web Survey System in your organization.

Designing Forms for the Web

The Epi Info™ Web Survey System allows for publishing short, single form projects (without grids or related forms) to the web for online data entry. This design creates certain features and behavioral differences between the web survey format and the full Epi Info™ functionality. Be aware of these differences and plan for them when designing a survey. For example, not all field types or check code commands are supported or provide value on a web survey. Additionally, required fields and drop-down lists may behave differently on a browser than they do on the Epi Info™ 7 Enter tool.

Before creating a survey, set the computer resolution to 1366 x 768 or 1024 x 768. These are the two most common screen resolutions in use today according to GS.StatCounter.com. If participants take the survey on an average non-mobile browser, this adjustment aligns the resolution to that of the most common participant. This will mitigate any discrepancies between how the survey appears to the designer and the participant.

These specifications are not necessary for participants using browsers on smart phones, tablets or other mobile devices. The Epi Info™ Web Survey System automatically adjusts the display when a survey is accessed with a mobile device. Surveys for these small screensized devices appear with a stacked question arrangement by tab order. For small mobile devices, horizontal field placement is irrelevant, as the fields appear one above the next.

After creating a new blank project, but before adding fields to the page, consider the page layout and size. A 'landscape' page orientation provides more width than height, which is consistent with most web pages. To set the page size and orientation, select **Format > Page Setup** from the menu. On the **Size** drop-down list, select the **size** to use or select **Custom** and enter the width and height in pixels. On the Orientation group, select the **Landscape** option.

Supported Field Types

The Web Survey System does not support all field types available in the Form Designer tool. See below for a list of supported and non-supported field types.

Supported field types:

- Label/Title
- Text (single line)
- Multiline (text)
- Number
- Date
- Time (not "Date/Time")
- Checkbox
- Yes/No
- Option
- Legal Values
- Comment Legal
- Codes
- Group

Non-Supported field types:

- Text (Uppercase)
- Unique Identifier
- Phone Number
- Date/Time
- Command Button
- Image
- Mirror
- Grid
- Relate

If a survey contains one or more of the non-supported field types, a message will identify the field(s) that will not publish. To publish the survey, perform one of the following two actions:

1. Delete the unsupported field(s). This will also delete any data that may have been in the Epi Info™ 7 Enter tool. If you want to preserve pre-existing data, make a project template and create a new project that you will customize for web publishing. In this new project, delete the non-supported fields.
2. If a data table does not exist for the form, you can use the "Change To" feature to change the field type to one of the supported types. For example, you can change an unsupported Text (Uppercase) field to a standard Text field or a Multiline field.

Supported Check Code Commands and Functions

The Web Survey tool does not support all Check Code commands and functions available in the Form Designer tool. See below for a list of supported and non-supported commands and functions.

Supported Commands:

- Assign
- Clear
- Dialog
    - Simple Dialog only
- Enable / Disable,
- Go To
- Hide / Unhide
- Highlight / Unhighlight
- If / Then / Else
- Set-Required / Set-Not-Required

Non-Supported Commands:

- AutoSearch
- Define
- Dialog
    - Get Variable Dialog
    - List of Values Dialog
- Execute
- Geocode
- Help
- NewRecord
- Quit

Supported Functions:

- ABS
- Cos
- Day
- Days
- FindText
- Hour
- Hours
- LN
- LOG
- Minute
- Minutes
- Month
- Months
- Rnd
- Round
- Second
- Seconds
- Sin
- StrLen
- Substring
- SystemDate
- SystemTime
- Tan
- Trunc
- UpperCase
- Year
- Years

Non-Supported Functions:

- CurrentUser
- DateDiff
- Date
- Environ
- EpiWeek
- Exists
- Exp
- FileDate
- Format
- Linebreak
- NumToDate
- NumToTime
- PFROMZ
- RecordCount
- Step
- TxtToDate
- TxtToNum
- ZSCORE

Although the non-supported commands and functions do not work when entering data into the web survey, they are still compatible with Check Code on the downloaded version of the survey. After downloading the web survey responses, you can run these commands and functions as normal, or you can use Epi Info™ 7 Analysis tool to perform the necessary operation.

Differences in Check Code Performance

In Check Code, the designer places commands within field blocks for actions such as 'Before', 'After', and for some field types, 'Click'. The key variation in performance between Check Code in the Enter tool and the Web Survey is related to the After block for fields that have drop-down lists (such as Yes/No, Legal Values, Comment Legal, and Codes).

In Enter, Check Code in the After block for fields executes when you leave the field either by tabbing to the next field or by clicking on another field on the page. However, in the Web Survey System, the Check Code in the After block for these fields executes as soon as you make a selection. The variation occurs since navigation is typically done by mouse and without a Tab key in the Web Survey System. Therefore, any Check Code commands (i.e. Enable/Disable, Hide/Unhide, Highlight/Unhighlight, and Set-Required/Set-Not-Required, etc.) that depend on the value selected in the drop-down list will occur as soon as the selection is made.

The Required property requires the participant to enter a value for the specified field. In the Enter tool, required fields must contain a value for the respondent to save the record or navigate to a different record. In the Web Survey tool, access to a separate page is also denied if a required field is left blank. If a required field is blank, a message displays notifying the participant that values must be given for required fields before the response can be submitted.

Confirm Web Survey Compatibility

To ensure the web survey functions properly, check the tab order and enter test data into the survey. Additionally, practice sample analyses on the test data before publishing the form.

A logical tab order is critical, especially if the participant might use a mobile device like a smart phone or tablet computer. As previously mentioned, tab order determines the order in which the fields are displayed on the mobile device. Even if the survey participant will use a desktop or laptop, any Check Code used to validate the data entry may depend on a logical tab order. For information regarding Tab Order, refer to the Form Designer section – How To: Set Tab Order.

Review all field names to ensure each is concise and logical. Field names cannot be changed after publishing the final version. A user-friendly way to check the field names is with the Data Dictionary feature. For information regarding Data Dictionary, refer to the Form Designer section – How To: View a Data Dictionary.

With the tab order and field names confirmed, enter sample test records. Click the **Enter Data** button from the Form Designer menu bar to preview the form in the Enter tool. When entering test data, record any changes that need to be made prior to publishing the survey. Check the field validation to determine if the field should be required, or if there is a valid range for number and date fields, or any other Check Code that may be used.

With test data in the form, attempt the analyses you anticipate performing. Confirm the field names are clear and easy to use. If you intend to use a tool other than Epi Info™ 7 for analysis, export the test data in the desired format. Common corrections include resetting a long field name, or changing/adding a field type.

Note: Corrections can only be made before the survey is distributed to participants for data collection.

Next, practice publishing the survey as a test by following the Publish Form to Web instructions. Confirm that you can successfully publish the form, navigate among the pages, enter more test data, and reconfirm the validations and Check Code behavior is all working properly. To ensure full functionality, download the test data and perform a sample analysis. If an error is observed, make the necessary corrections and repeat this step until the survey functions as desired.

When testing is complete, remove all test data using the Delete Data Table function and publish the final version. For information regarding Delete Data Table, refer to the Form Designer section – How To: Delete an Existing Data Table Without Deleting the Form.

Initial Setup

Publishing a survey and collecting data requires information that is either provided by the Web Survey System administrator or generated by the Web Survey System. This information provides security to the system and access to individual surveys. The following are key terms used in Web Survey:

- **Endpoint Address** – designates the location or URL of the application that hosts surveys.
- **Organization Key** – provides survey designer access to publish the survey and import the responses.
- **Survey Key** – identifies the survey that corresponds to the Security Token for downloading data.
- **Security Token** – grants access to the survey data and prevents others with the Organization Key and Survey Key from accessing the data**.**

Before publishing a survey, the designer must designate the endpoint address and binding protocol.

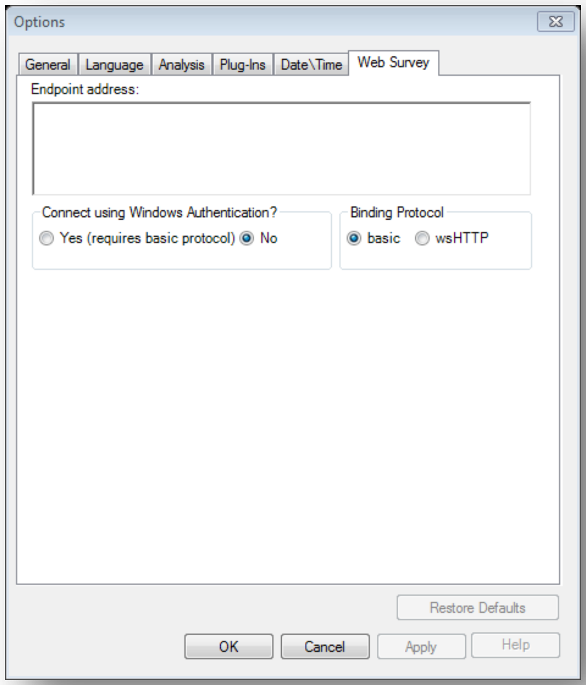
12. Contact your Web Survey System administrator for the following:

    - Endpoint Address

- Use of Windows Authentication (Yes or No)
- Binding Protocol (basic or wsHTTP)
- Organization Key

13. From the Epi Info™ Form Designer menu, select **Tools > Options.**

14. Click on the **Web Survey** tab. The **Web Survey Options** dialog box appears.



Endpoint address:

Connect using Windows Authentication? Yes (requires basic protocol) [✓ No]

Binding Protocol [✓ basic] wsHTTP

Figure 7.219: Web Survey Options dialog box

15. Enter the **Endpoint address**, and then make your **Windows Authentication**, and **Binding Protocol** selections. The Web Survey System administrator provides this information. An example of an endpoint address is: http://MySurveyManager.MyCompany.com/SurveyManagerService.svc

Publish Form to Web

After designing a form that is compatible with the Web Survey System, the survey can be published to the web server. With the completed form open in Form Designer, complete the following steps.

2. From the Form Designer menu, select **File > Publish Form to Web.**

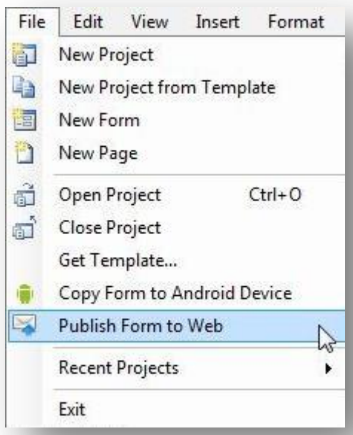
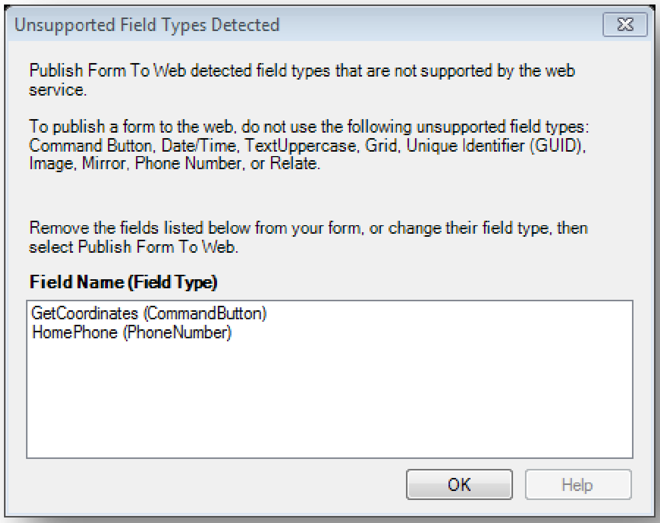


Figure 7.220 Publish Form to Web option

3. If a form contains a non-supported field type, an error message occurs stating the name of each invalid field.


Unsupported Field Types Detected

Publish Form To Web detected field types that are not supported by the web service.

To publish a form to the web, do not use the following unsupported field types: Command Button, Date/Time, TextUppercase, Grid, Unique Identifier (GUID), Image, Mirror, Phone Number, or Relate.

Remove the fields listed below from your form, or change their field type, then select Publish Form To Web.

Field Name (Field Type)

GetCoordinates (CommandButton)
HomePhone (PhoneNumber)

Figure 7.221: Unsupported Field Types error message

To meet publishing requirements, delete or change the field type of non-supported fields. Before deleting a field, be aware that any data collected with that field will be deleted as well. The data removal is automatic and irreversible. Consider backing up all data prior to deleting a field. Alternatively, use the **Change To** option to change the field to a supported field type. This action can only be performed if there is not a data table associated with the form. For information regarding Delete Data Table, refer to the Form Designer section – How To: Delete an Existing Data Table Without Deleting the Form.

Note: Data tables are created when the form is accessed with the Enter Data tool, regardless of whether data has been collected. To remove a data table, select **Tools>Delete Data Table**.

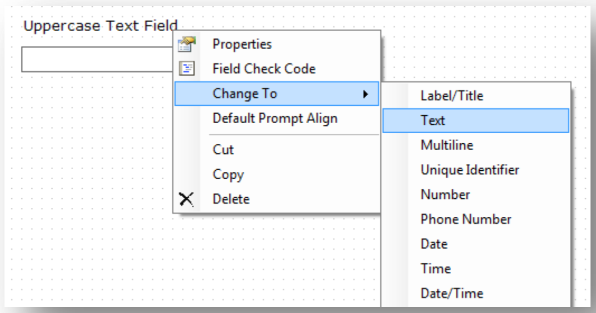


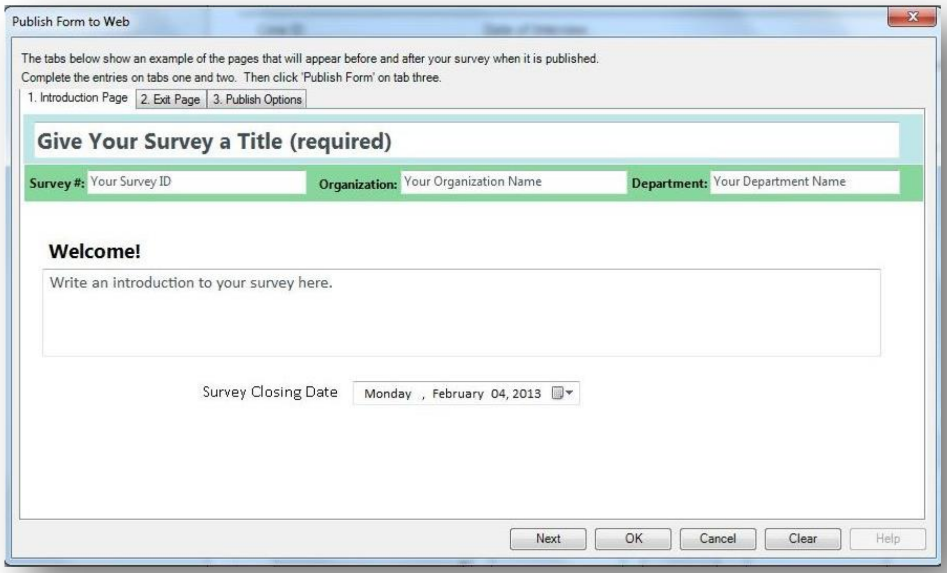
Figure 7.222: Fields Change To function

4. In the Publish Form to Web dialog box, there are three tabs that must be filled out prior to publishing the survey:

    - **Introduction Page** – provides information to the survey participant prior to beginning the survey
    - **Exit Page** – provides closing remarks to the survey participant after the survey is submitted
    - **Publish Options** – specifies technical aspects of the web survey posting process

The following navigation buttons are at the bottom of the dialog box:

- The **Next** button saves the information and proceeds to the next tab.
- The **Clear** button removes all entries from the page and returns the user to a blank Introduction Page.
- The **Cancel** button allows the user to exit the Publish Form to Web dialog box and return to the Form Designer window.
- The **OK** button will attempt to publish the survey. If any information is omitted, an error message will appear.


The tabs below show an example of the pages that will appear before and after your survey when it is published.
Complete the entries on tabs one and two. Then click 'Publish Form' on tab three.

Publish Form to Web
1. Introduction Page
2. Exit Page
3. Publish Options
Give Your Survey a Title (required)
Survey #: Your Survey ID
Organization: Your Organization Name
Department: Your Department Name
Welcome!
Write an introduction to your survey here.
Survey Closing Date
Monday , February 04, 2013
Next
OK
Cancel
Clear
Help

Figure 7.223: Publish Form to Web dialog box

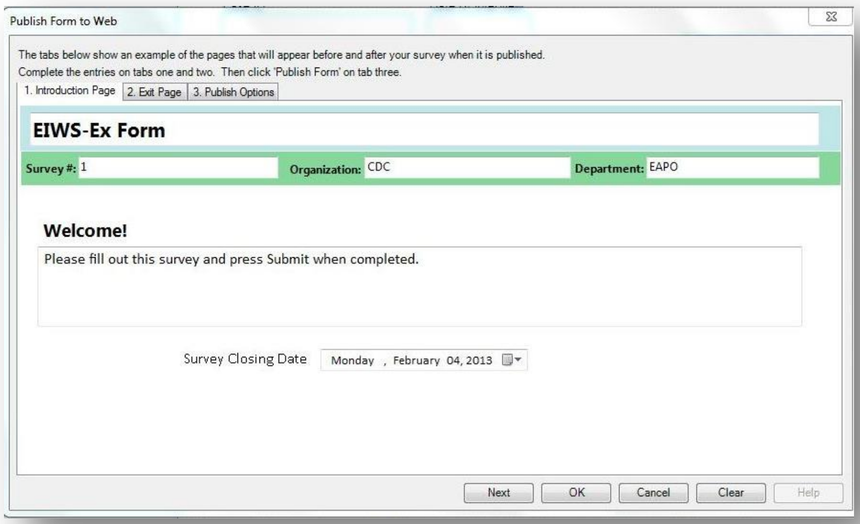
4. On the Introduction Page tab, click in the **Give Your Survey a Title (required)** textbox to enter the survey title. This title will appear at the top of each page in the survey.

5. Enter your **Survey Number or ID, Organization Name** and **Department Name**. This information appears at the top of the introduction and exit pages.

Note: The Organization Name is used for tracking purposes only.
The Organization Key is the key that allows you to post a survey to the web server.
The Organization Key should be kept secure and confidential.

DO NOT PUT THE ORGANIZATION KEY INTO THE 'Organization Name' FIELD ON THE INTRODUCTION TAB.

6. The **Welcome!** textbox provides participants with a greeting, special instructions or additional information about the survey. Click in the **Welcome!** textbox and enter a message.


Publish Form to Web

The tabs below show an example of the pages that will appear before and after your survey when it is published.
Complete the entries on tabs one and two. Then click 'Publish Form' on tab three.

EIWS-Ex Form

Survey #: 1 Organization: CDC Department: EAPO

Welcome!

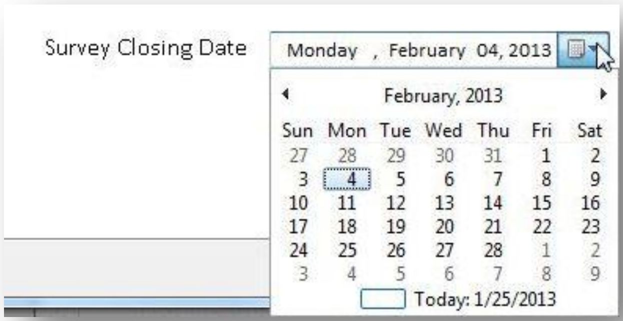
Please fill out this survey and press Submit when completed.

Survey Closing Date Monday , February 04, 2013

Figure 7.224: Introduction Page

7. Select the survey closing date from the **Survey Closing Date** drop-down menu located below the **Welcome!** textbox. This field only accepts a date in the future.

Note: The default setting is 10 days from the current date.


Survey Closing Date Monday , February 04, 2013

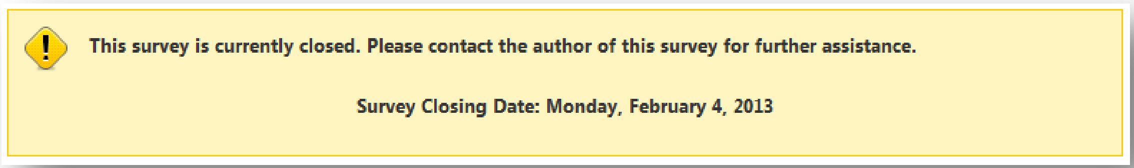
February, 2013

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	1	2
3	4	5	6	7	8	9

Today: 1/25/2013

Figure 7.225: Survey Closing Date

At 11:59:59.99 pm on the selected survey closing date, any attempt to access the survey will prompt the message below.

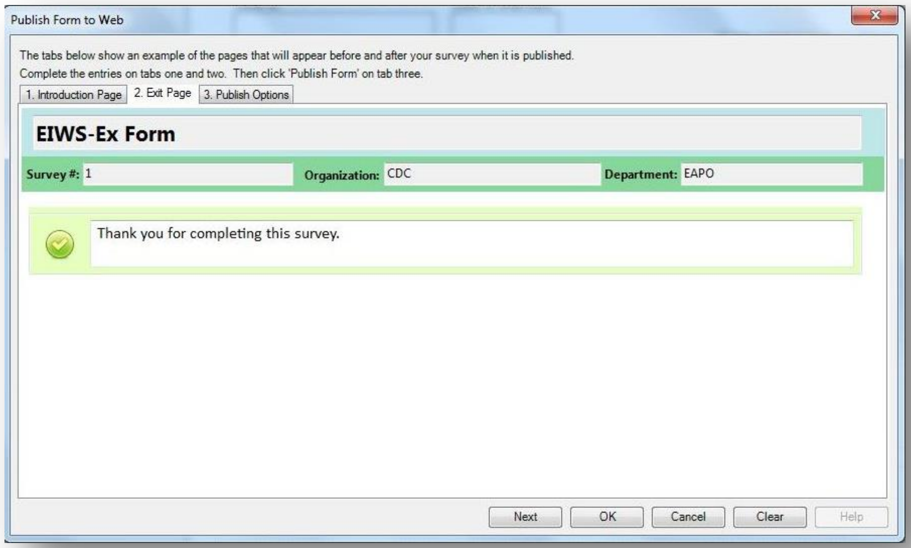

This survey is currently closed. Please contact the author of this survey for further assistance.

Survey Closing Date: Monday, February 4, 2013

Figure 7.226: Closed Survey dialog box

8. Click **Next** to save this information and proceed to the Exit Page tab.

Note: The Introduction Page can be edited at any point prior to publishing the survey by clicking on the Introduction Page tab.



The tabs below show an example of the pages that will appear before and after your survey when it is published.
Complete the entries on tabs one and two. Then click 'Publish Form' on tab three.

EIWS-Ex Form

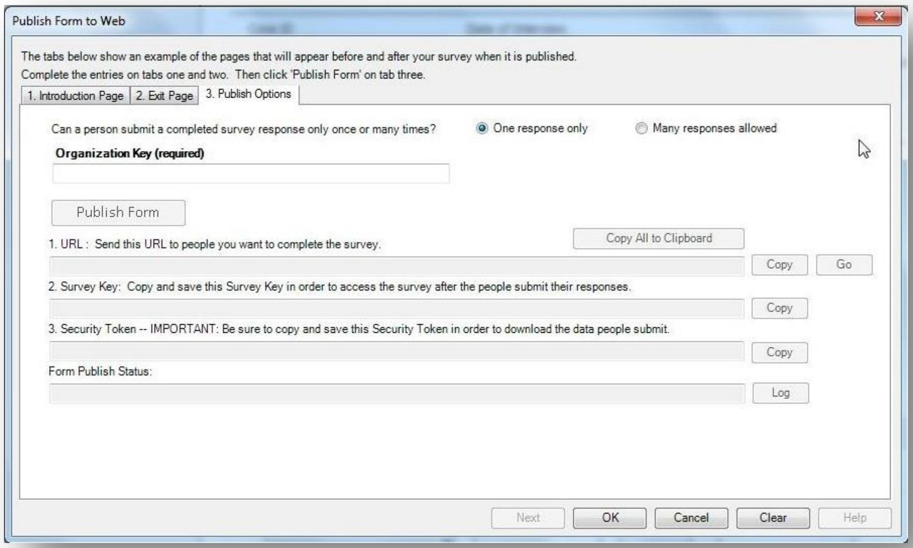
Survey #: 1 Organization: CDC Department: EAPO

Thank you for completing this survey.

Figure 7.227: Exit Page

9. The Exit Page displays closing remarks or additional instructions about the survey. Enter your comments in the textbox next to the green check mark. This example states "Thank you for completing this survey." The Survey #, Organization and Department fields will automatically populate based on the information provided in the Introduction Page tab. Clicking on either of these fields will direct you back to the Introduction Page.

10. Click **Next** to proceed to the Publish Options tab.


Publish Form to Web

The tabs below show an example of the pages that will appear before and after your survey when it is published.
Complete the entries on tabs one and two. Then click 'Publish Form' on tab three.

Can a person submit a completed survey response only once or many times? [✓ One response only] Many responses allowed

Organization Key (required)

1. URL : Send this URL to people you want to complete the survey.

2. Survey Key: Copy and save this Survey Key in order to access the survey after the people submit their responses.

3. Security Token -- IMPORTANT: Be sure to copy and save this Security Token in order to download the data people submit.

Form Publish Status:

Figure 7.228: Publish Options

11. To allow a participant to submit more than one response in a given session, select the **Many responses allowed** option button. The default setting is **One response only,** which closes the survey after the participant submits one record.

Note: Selecting "*One response only*" will not prevent a participant from accessing a survey link more than once.

12. Enter/Paste the **Organization Key** provided by the Web Survey System administrator.

Hint: If the organization key is provided in an email or other electronic file, copy and paste the key into the organization key field to avoid any mistakes. The organization key must be a valid global unique identifier to enable the Publish Form button. An example of an Organization Key is "3f6c6251-4af6-40f7-a440-cb2a9ca92e17."

13. Click **Publish Form**.

If an error message appears, contact your Web Survey System administrator with the text displayed in the **Form Publish Status** field. Click the associated **Log** button to access detailed information regarding the error message if available. Typically, this error occurs with an invalid or missing endpoint address.

After successfully publishing the survey, the **URL**, **Survey Key** and **Security Token** fields will automatically populate. Survey participants will use this URL to access the survey.

14. Click the **Copy All to Clipboard** button. Save this information in a document or a format of your choosing for future use.

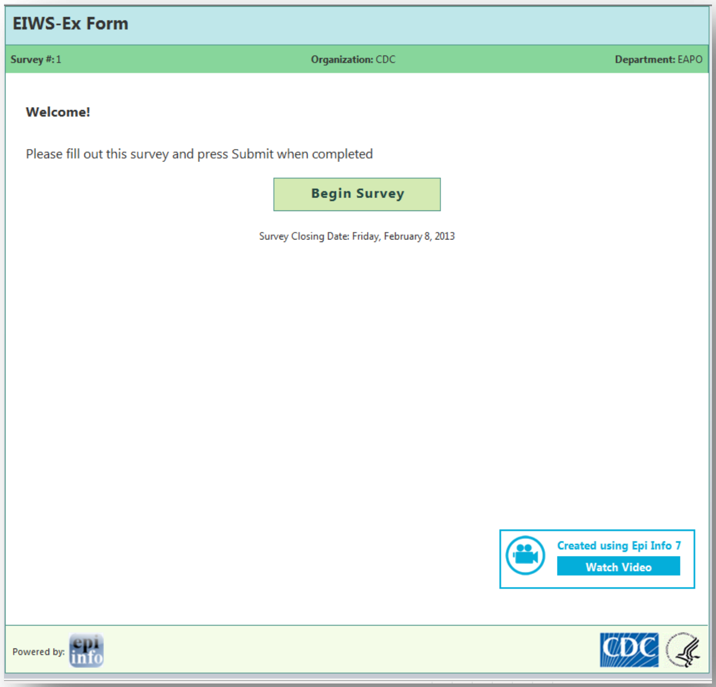
Important: The URL, survey key and security token are all required to access the survey and survey data. The security token cannot be recreated if it is lost. Keep the security token in a secure location. DO NOT SHARE THE SECURITY TOKEN WITH SURVEY PARTICIPANTS.

Invite Participants to Take the Survey

After publishing the survey, participants can access the web survey at any time up to and including the survey closing date. The survey can be accessed using the URL generated in the Publish Form to Web steps in the previous section. Designers can invite participants by distributing the web address or URL over email, by posting it on a web page, or using other methods.

Access the Survey Using the URL

4. Launch your web browser. Copy and paste the **URL** into the address bar at the top of your browser window. If the survey is not closed, the survey Welcome! page appears.


EIWS-Ex Form

Survey #: 1 Organization: CDC Department: EAPO

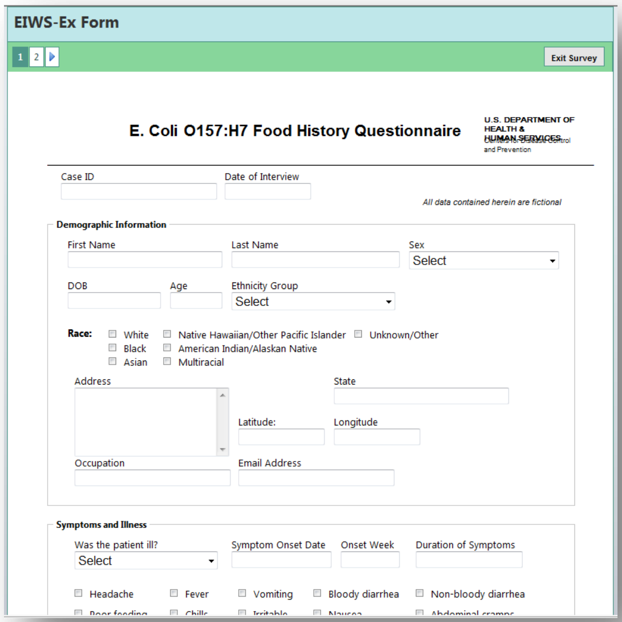
Welcome!

Please fill out this survey and press Submit when completed

Survey Closing Date: Friday, February 8, 2013

Figure 7.229: Survey Welcome Page

5. Click **Begin Survey.** The first page of the survey appears.



EIWS-Ex Form

1 2 ▸ Exit Survey

E. Coli O157:H7 Food History Questionnaire

U.S. DEPARTMENT OF HEALTH & HUMAN SERVICES

Case ID | Date of Interview

All data contained herein are fictional

Demographic Information

First Name | Last Name | Sex: Select

DOB | Age | Ethnicity Group: Select

Race: ☐ White ☐ Black ☐ Asian ☐ Native Hawaiian/Other Pacific Islander ☐ American Indian/Alaskan Native ☐ Multiracial ☐ Unknown/Other

Address | State | Latitude: | Longitude

Occupation | Email Address

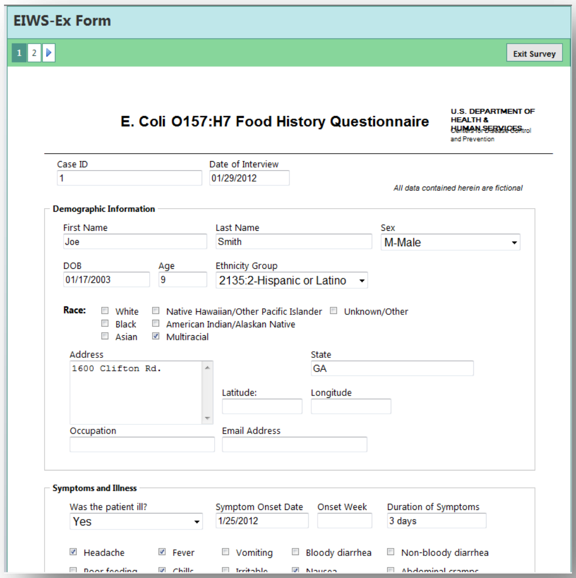
Symptoms and Illness

Was the patient ill?: Select | Symptom Onset Date | Onset Week | Duration of Symptoms

☐ Headache ☐ Fever ☐ Vomiting ☐ Bloody diarrhea ☐ Non-bloody diarrhea

Figure 7.230: Published Web Survey form

Enter your response to each survey question. You can go to the next page of the survey by clicking on the navigation buttons on the top left corner of the screen. You can also click the **Continue** button at the bottom of the page to navigate between pages. The information entered into the form is not saved until **Submit Survey** button or the **Finish Later** button at the bottom of the page is clicked. For information regarding the Finish Later button, refer to the Saving Survey Answers for Future Submission topic.



EIWS-Ex Form

1 2 ▸ Exit Survey

E. Coli O157:H7 Food History Questionnaire

U.S. DEPARTMENT OF HEALTH & HUMAN SERVICES

Case ID: 1 | Date of Interview: 01/29/2012

All data contained herein are fictional

Demographic Information

First Name: Joe | Last Name: Smith | Sex: M-Male

DOB: 01/17/2003 | Age: 9 | Ethnicity Group: 2135:2-Hispanic or Latino

Race: ☐ White ☐ Black ☐ Asian ☐ Native Hawaiian/Other Pacific Islander ☐ American Indian/Alaskan Native ☑ Multiracial ☐ Unknown/Other

Address: 1600 Clifton Rd. | State: GA | Latitude: | Longitude

Occupation | Email Address

Symptoms and Illness

Was the patient ill?: Yes | Symptom Onset Date: 1/25/2012 | Onset Week | Duration of Symptoms: 3 days

☑ Headache ☑ Fever ☐ Vomiting ☐ Bloody diarrhea ☐ Non-bloody diarrhea

Figure 7.231: E. coli Web Survey form, page 1

Submitting Survey Responses

1. After completing the last page of the survey, click **Submit Survey** to send the response.

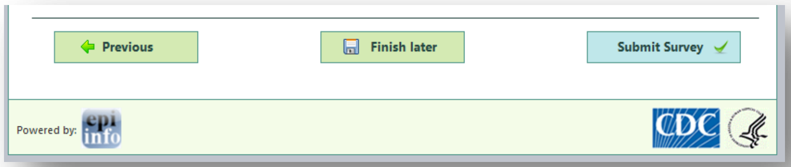
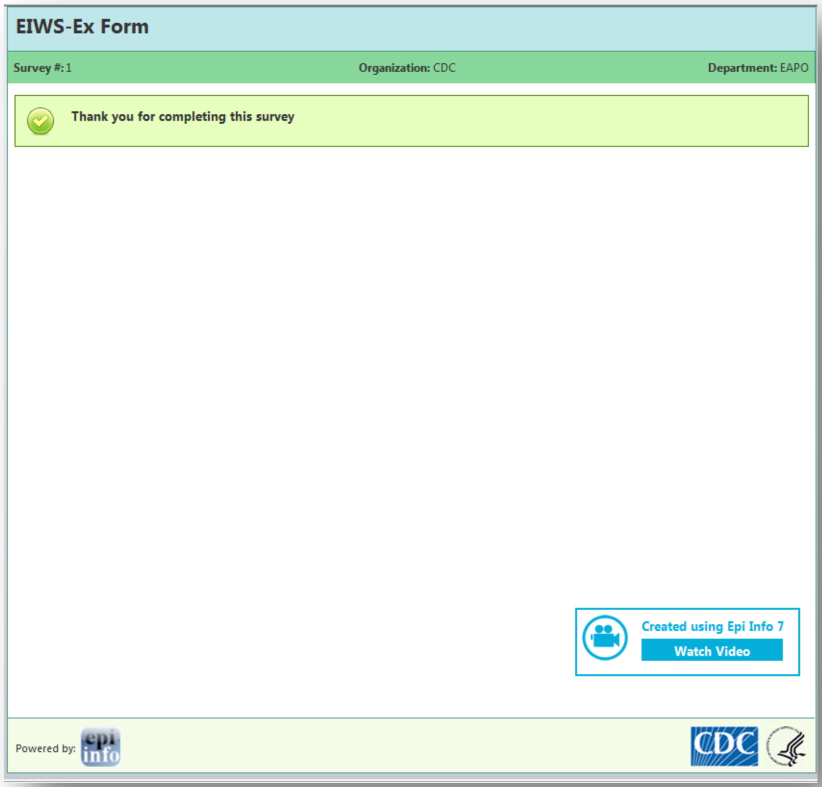


Figure 7.232: Submit Survey

2. The **Exit Page** appears and displays the thank you message.

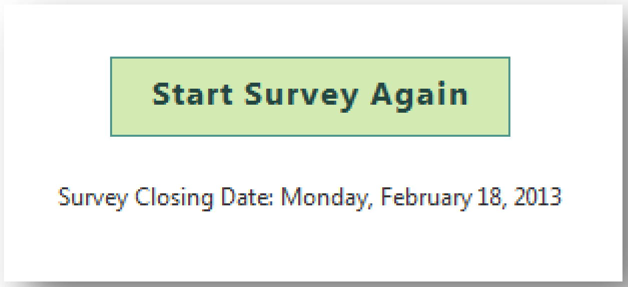

EIWS-Ex Form

Survey #: 1 Organization: CDC Department: EAPO

Thank you for completing this survey

Figure 7.233: Survey Exit Page

3. If the **Many responses allowed** option was selected in the Publish Options tab, the Start Survey Again message will appear allowing the participant to submit more than one response. This is useful when one participant is entering multiple responses to the survey.



Survey Closing Date: Monday, February 18, 2013

Figure 7.234: Start Survey Again

Saving Survey Answers for Future Submission

If the survey cannot be completed in one sitting, participants can save the survey and finish their responses at a later time.

1. To save a survey, click the green **Finish later** button located at the bottom of each page in the survey.

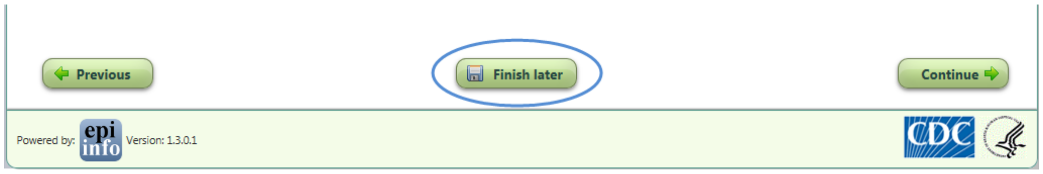
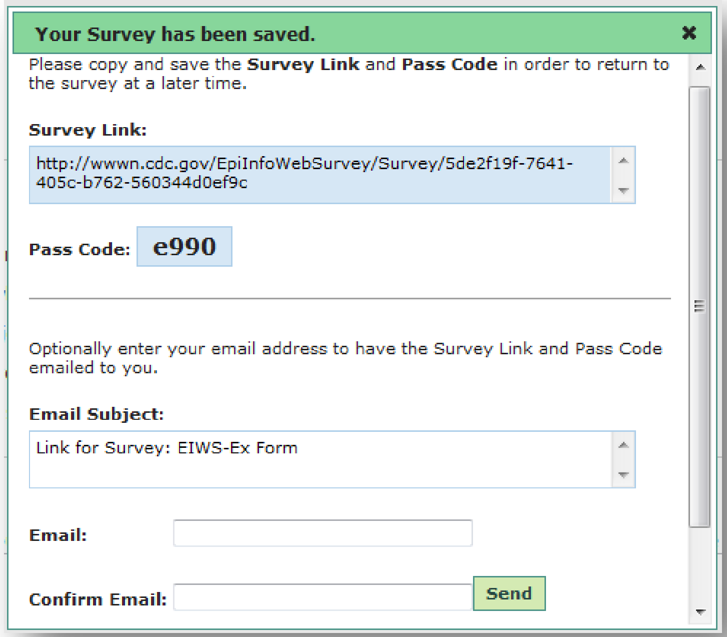


Figure 7.235: Finish Survey later

A dialog box appears with a Survey Link and Pass Code.



Your Survey has been saved.

Please copy and save the **Survey Link** and **Pass Code** in order to return to the survey at a later time.

Survey Link:

http://wwwn.cdc.gov/EpiInfoWebSurvey/Survey/5de2f19f-7641-405c-b762-560344d0ef9c

Pass Code: e990

Optionally enter your email address to have the Survey Link and Pass Code emailed to you.

Email Subject:

Link for Survey: EIWS-Ex Form

Email:

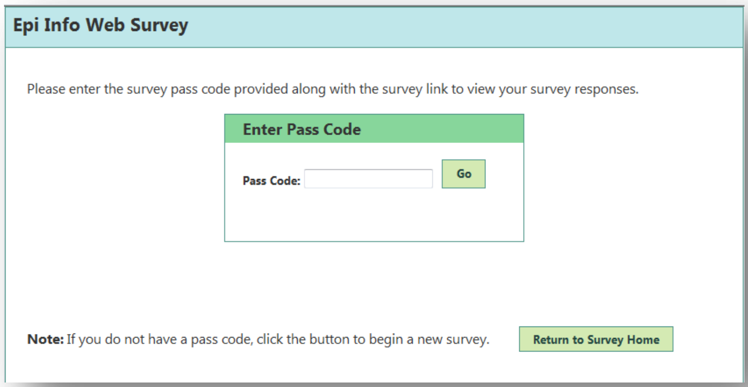
Confirm Email:

Figure 7.236: Survey Link and Pass Code

2. Copy and save the **Survey Link** and **Pass Code.** This information is required to return to the survey at a later time. The saved survey contains all information entered into the form and cannot be retrieved by the participant without the Survey Link and Pass Code. If the participant loses the Pass Code or Survey Link, the participant will need to take the survey from the beginning.

There is an option to email the Survey Link and Pass Code. The email address is not saved in the system and is only used to generate the outgoing email.

3. Enter a valid **email address** in the **Email** textbox.
4. Re-enter the **email address** in the **Confirm Email** textbox.
5. Click **Send**. The participant will receive the Survey Link and Pass Code via email.
6. To complete the survey, copy and paste the **Survey Link** into a web browser or click the link in the email.
7. Enter the Pass Code into the **Pass Code** textbox. Click **Go**.



Epi Info Web Survey

Please enter the survey pass code provided along with the survey link to view your survey responses.

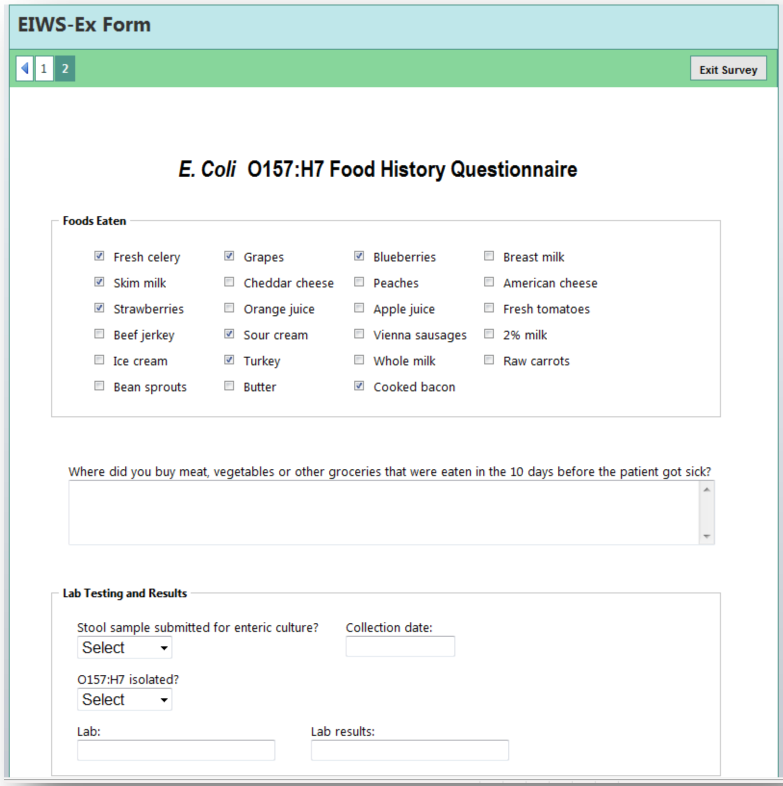
Enter Pass Code

Pass Code: [Go]

Note: If you do not have a pass code, click the button to begin a new survey. [Return to Survey Home]

Figure 7.237: Enter Pass Code to complete survey

The saved survey form appears containing all information previously entered.



EIWS-Ex Form

1 2 Exit Survey

E. Coli O157:H7 Food History Questionnaire

Foods Eaten

☑ Fresh celery	☑ Grapes	☑ Blueberries	☐ Breast milk
☑ Skim milk	☐ Cheddar cheese	☐ Peaches	☐ American cheese
☑ Strawberries	☐ Orange juice	☐ Apple juice	☐ Fresh tomatoes
☐ Beef jerkey	☑ Sour cream	☐ Vienna sausages	☐ 2% milk
☐ Ice cream	☑ Turkey	☐ Whole milk	☐ Raw carrots
☐ Bean sprouts	☐ Butter	☑ Cooked bacon	

Where did you buy meat, vegetables or other groceries that were eaten in the 10 days before the patient got sick?

Lab Testing and Results

Stool sample submitted for enteric culture? Select

Collection date:

O157:H7 isolated? Select

Lab:

Lab results:

Figure 7.238: Partially Completed Web Survey Form

Importing Survey Data

1. From the Epi Info™ main menu, select **Enter Data** or select **Tools > Enter Data**. The Enter tool opens.
2. Open the project and form used to publish the web survey.
3. From the File menu, select **Import Data > From Web**.

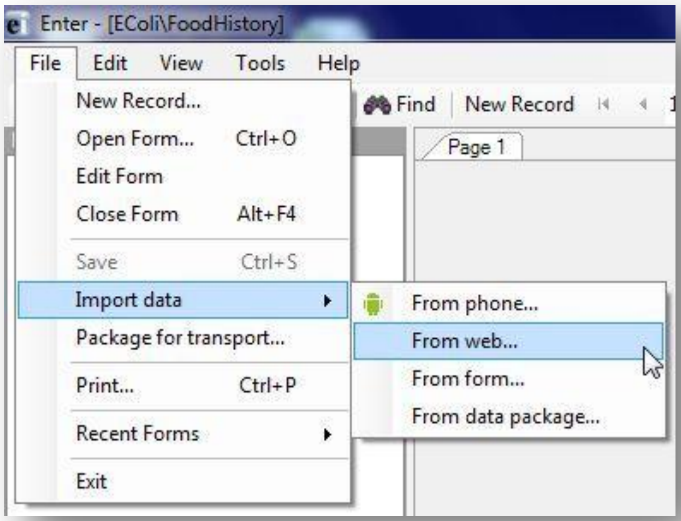
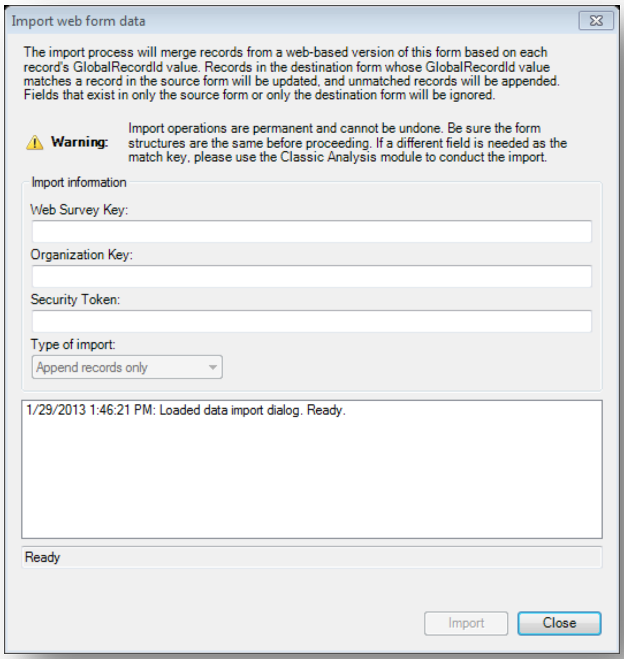


Figure 7.239: Import Data from Web

4. Enter the **Web Survey Key**, **Organization Key**, and **Security Token** into the **Import web form data** dialog box (Refer to Publish Form to Web, Step 13 for this information.)


Import web form data

The import process will merge records from a web-based version of this form based on each record's GlobalRecordId value. Records in the destination form whose GlobalRecordId value matches a record in the source form will be updated, and unmatched records will be appended. Fields that exist in only the source form or only the destination form will be ignored.

Warning: Import operations are permanent and cannot be undone. Be sure the form structures are the same before proceeding. If a different field is needed as the match key, please use the Classic Analysis module to conduct the import.

Import information

Web Survey Key:

Organization Key:

Security Token:

Type of import: Append records only

```
1/29/2013 1:46:21 PM: Loaded data import dialog. Ready.
```

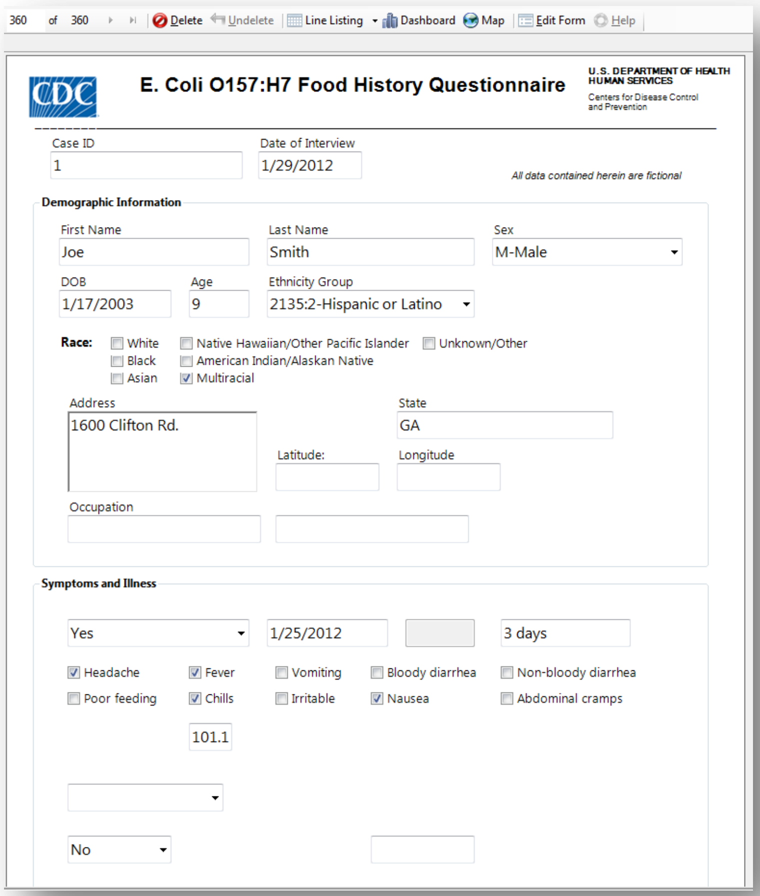
Ready

Figure 7.240: Import Web Form Data dialog box

5. Click **Import.**

Note: If you incur an error message, ensure that the endpoint address is entered in the Web Survey dialog box.

6. After the import is complete, click **Close**. The responses to the survey questions appear in the survey form and are added to your Epi Info™ database. The import process can be executed multiple times and will append newly submitted responses without creating duplicate records.


360 of 360
Delete
Undelete
Line Listing
Dashboard
Map
Edit Form
Help
U.S. DEPARTMENT OF HEALTH HUMAN SERVICES
Centers for Disease Control and Prevention

E. Coli O157:H7 Food History Questionnaire

Case ID: 1
Date of Interview: 1/29/2012

All data contained herein are fictional

Demographic Information

First Name: Joe
Last Name: Smith
Sex: M-Male
DOB: 1/17/2003
Age: 9
Ethnicity Group: 2135:2-Hispanic or Latino

Race: White ☐ Black ☐ Asian ☐ Native Hawaiian/Other Pacific Islander ☐ American Indian/Alaskan Native ☐ Multiracial ☑ Unknown/Other ☐

Address: 1600 Clifton Rd.
State: GA
Latitude:
Longitude
Occupation

Symptoms and Illness

Yes | 1/25/2012 | 3 days

Headache ☑ Fever ☑ Vomiting ☐ Bloody diarrhea ☐ Non-bloody diarrhea ☐
Poor feeding ☐ Chills ☑ Irritable ☐ Nausea ☑ Abdominal cramps ☐

101.1

No

Figure 7.241: Imported Data

The data collected is now ready for analysis. Refer to the Visual Dashboard or Classic Analysis sections of this user guide for further information about analysis.

6. Companion for Android: Collecting and Analyzing Data on an Android Device

Introduction

Epi Info™ Companion for Android allows users to transfer forms to a mobile device and collect data remotely. The mobile application is compatible with Android tablets and phones running Android version 4.0 (codename "Ice Cream Sandwich") and above. The ability to load Epi Info™ forms onto a mobile device allows users to collect data in places lacking information technology infrastructure and for activities that could benefit from mobility, GPS tacking, or photographic capabilities. Once data is collected, users can perform simple analysis with the Analyze Data function or view geographic data using the built in mapping function. These tools do not contain the full Epi Info™ functionality; however, data can be transferred from the mobile device to a PC for more complex analyses. The mobile application also contains a full featured StatCalc function similar to the one found in Epi Info™ 7.

Supported Field Types

Epi Info™ Companion for Android does not support all field types available in Form Designer. See below for a list of supported and non-supported field types. When designing a form for use with the mobile application, exclude non-supported field types from the form.

Supported field types:

- Label/Title
- Text (single line)
- Text (Uppercase)
- Multiline (text)
- Number
- Date
- Time (not "Date/Time")
- Checkbox
- Yes/No
- Option
- Legal Values
- Comment Legal
- Group
- Command Button
- Image

Non-Supported field types:

- Unique Identifier
- Phone Number
- Date/Time
- Mirror
- Grid
- Relate
- Codes

If a form contains one or more of the non-supported field types, a message will identify the field(s) that will not publish. To publish the form, perform one of the following two actions:

3. Delete the unsupported field(s). This will also delete any data that may have been in the Epi Info™ 7 Enter tool. If you want to preserve pre-existing data, make a project template and create a new project that you will customize for the Mobile application. In this new project, delete the non-supported fields.
4. If a data table does not exist for the form, you can use the "Change To" feature to change the field type to one of the supported types. For example, you can change an unsupported Text (Uppercase) field to a standard Text field or a Multiline field.

Supported Check Code Commands and Functions

Epi Info™ Companion for Android does not support all check code commands and functions available in the Form Designer tool. See below for a list of supported and non-supported commands and functions. When designing a form for use with the mobile application, exclude non-supported check code commands from the form.

Supported Commands:

- Assign
- Clear
- Dialog
    - Simple Dialog only
- Enable / Disable,
- Go To
- Hide / Unhide
- Highlight / Unhighlight
- If / Then / Else
- Set-Required / Set-Not-Required

Non-Supported Commands:

- AutoSearch
- Define
- Dialog
    - Get Variable Dialog
    - List of Values Dialog
- Execute
- Geocode
- Help
- NewRecord
- Quit

Supported Functions:

- ABS
- Cos
- Day
- Days
- FindText
- Hour
- Hours
- LN
- LOG
- Minute
- Minutes
- Month
- Months
- Rnd
- Round
- Second
- Seconds
- Sin
- StrLen
- Substring
- SystemDate
- SystemTime
- Tan
- Trunc
- UpperCase
- Year
- Years

Non-Supported Functions:

- CurrentUser
- DateDiff
- Date
- Environ
- EpiWeek
- Exists
- Exp
- FileDate
- Format
- Linebreak
- NumToDate
- NumToTime
- PFROMZ
- RecordCount
- Step
- TxtToDate
- TxtToNum
- ZSCORE

Although the non-supported commands and functions do not work when entering data into the mobile form, they are still compatible with check code on the downloaded version of the form. After downloading the responses to a PC, you can run these commands and functions as normal, or you can use Epi Info™ 7 Analysis y to perform the necessary operation.

Initial Setup

The Epi Info™ Companion for Android is available for download at the Google Play store. Epi Info Android Download Location (https://play.google.com/store/apps/details?id=gov.cdc.epiinfo&hl=en)

With the mobile device connected to the Internet, enter the web address into your browser and select **Download.** Connect the mobile device to your computer through a USB cable.

Note: The mobile device must be connected through a hard line and will not synchronize with your computer wirelessly.

The mobile application is compatible with Android versions 4.0 and above. Since Android specifications are open source, device manufacturers are free to change positioning and appearance of common elements such as menus and buttons. For example, users should be aware that Samsung Galaxy Note II devices have their "Back" button on the right hand side of the device and the "Options Menu" on the bottom, while Google Nexus 7 devices have their "Back" button on the left hand side of the device and the "Options Menu" on the top corner. Furthermore, when an Android device is connected to a PC via a USB cable, it may register as a disk drive, media device, or a digital camera. Please note that data and forms can only be transferred in disk drive or media device modes.

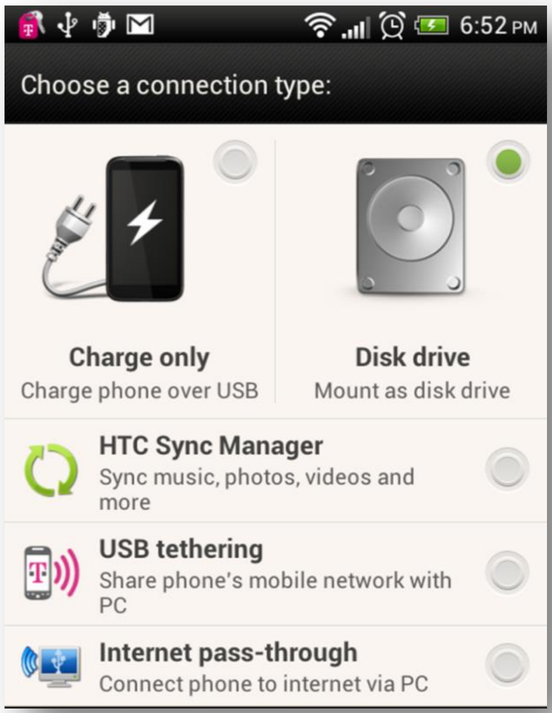
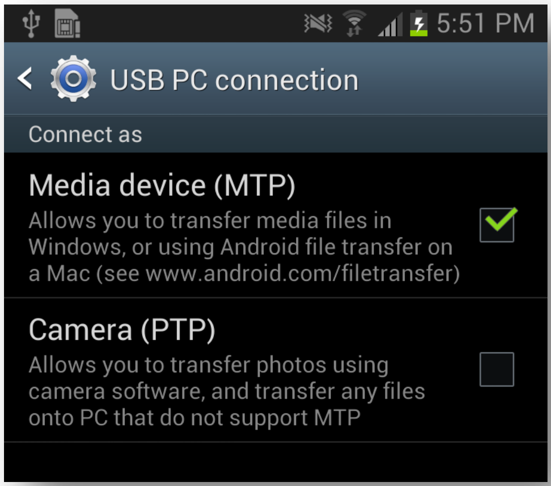

6:52 PM
Choose a connection type:
Charge only
Charge phone over USB
Disk drive
Mount as disk drive
HTC Sync Manager
Sync music, photos, videos and more
USB tethering
Share phone's mobile network with PC
Internet pass-through
Connect phone to internet via PC

Figure 7.242: Connection Type



Media device (MTP) ✓
Allows you to transfer media files in Windows, or using Android file transfer on a Mac (see www.android.com/filetransfer)

Camera (PTP)
Allows you to transfer photos using camera software, and transfer any files onto PC that do not support MTP

Figure 7.243: USB PC Connection

Internet connectivity is not required for using the core StatCalc, data collection, and data analysis modules. However, the mapping function utilizes Google Maps and requires Internet connectivity for downloading base map information.

Designing Forms for Mobile

Epi Info™ Companion for Android allows for the transferring of short, single form projects (without grids or related forms) to a mobile device for data entry. This design creates certain features and behavioral differences between the mobile and PC format. Be aware of these differences and plan for them when designing a form. For example, not all field types or check code commands are supported or provide value on a mobile device. Additionally, required fields and drop-down lists may behave differently on a mobile device than they do on the Epi Info™ 7 Enter tool.

Note: Forms on a mobile device require a minimum of three data fields.

Companion for Android automatically adjusts the display when a form is accessed with a mobile device. Forms on ten-inch tablets appear with a similar field layout as the PC. For smaller mobile devices (seven inches or less), horizontal field placement is irrelevant, as the fields are stacked vertically.

Note: Fields on a mobile phone display vertically according to the Tab Order. Set the tab order according to the sequence you want the fields to display on the form.

After uploading the form, conduct a usability test before collecting data. Ensure all fields are displayed properly and oriented in the intended order.

Copy Form to Mobile Device

From the Epi Info™ main menu, click **Create Forms** or select **Tools > Create Forms** from the navigation menu. Open the desired project form or create a new form to transfer to the mobile device. (See Form Designer).

5. Connect the mobile device to your computer via a USB cable. Ensure that the PC recognizes the mobile device in disk drive or media device mode before proceeding to the next step.

6. From the Form Designer menu, select **Copy Form to Android Device.**

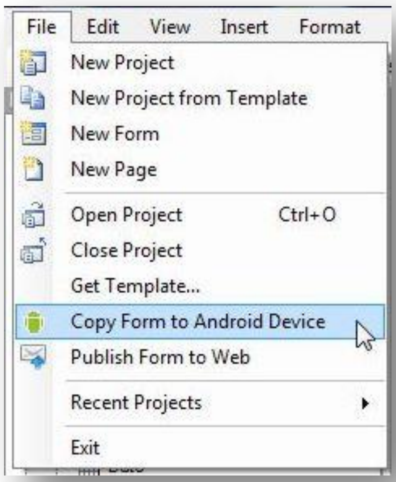
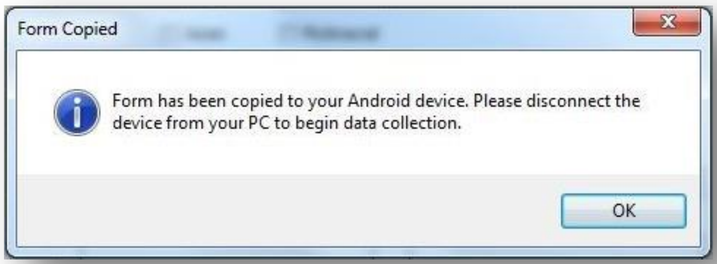


Figure 7.244: Copy Form to Android Device

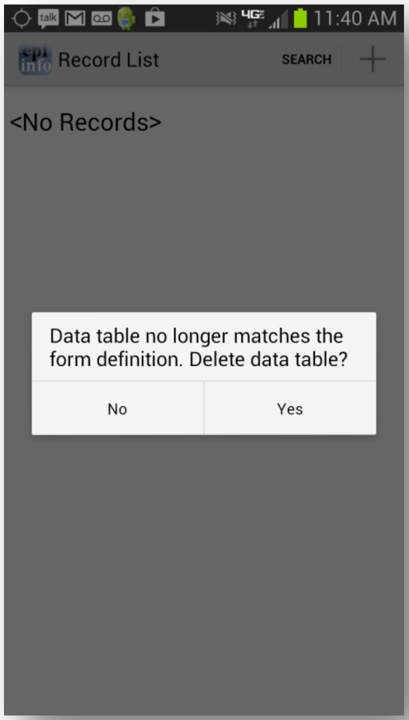
The form automatically transfers to the device and is ready for use.



Form has been copied to your Android device. Please disconnect the device from your PC to begin data collection.

Figure 7.245: Form Copied

When a form is uploaded to a mobile device to replace an existing one of the same name, an error may occur indicating that fields no longer match.

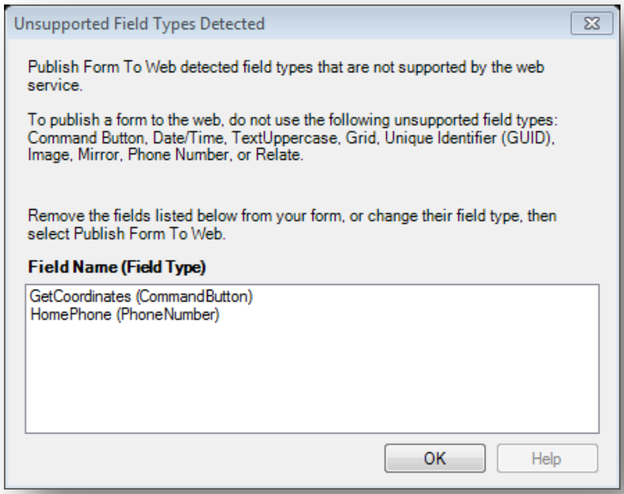


<No Records>

Data table no longer matches the form definition. Delete data table?

Figure 7.246: Data Table Error Message

Select **Yes** to delete the data table and continue data collection.



Publish Form To Web detected field types that are not supported by the web service.

To publish a form to the web, do not use the following unsupported field types: Command Button, Date/Time, TextUppercase, Grid, Unique Identifier (GUID), Image, Mirror, Phone Number, or Relate.

Remove the fields listed below from your form, or change their field type, then select Publish Form To Web.

Field Name (Field Type)

GetCoordinates (CommandButton)
HomePhone (PhoneNumber)

Figure 7.247: Unsupported Field Types error message

To meet publishing requirements, the non-supported fields must be deleted or the field type must be changed. Before deleting a field, be aware that any associated data with that field will be deleted as well. The data removal is automatic and irreversible. Consider backing up all data prior to deleting a field. Alternatively, use the “Change To” option to change the field to a supported field type. This action can only be performed if a data table is not associated with the form.

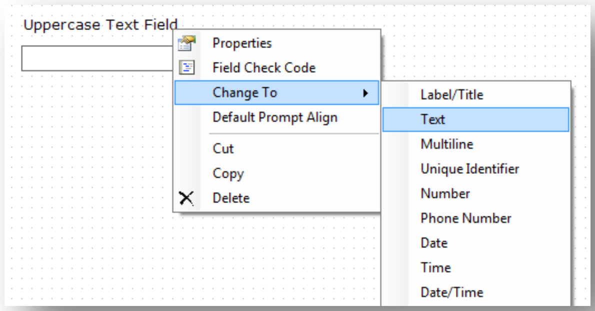


Figure 7.248: Fields Change To function

NOTE: If you are unable to copy the form into your device because the device is not recognized by Epi Info 7, you could copy the form into your device by creating a template of your form. In order to create a form template, please review the Form Designer chapter of the User’s Manual. Once the form template is created, you can simply copy the form template file (which will have an .XML file extension) into your device or email the form template to yourself or others. Once the file has been copied to your device, it would be recognized by the Companion for Android application and you will be able to begin data collection.

Collect Data on Mobile Device

Access the mobile companion titled Epi Info™ from your device’s application list. Note that the application layout is different between a tablet and a phone. The screen will appear in a portrait format on a phone and landscape format on a tablet.

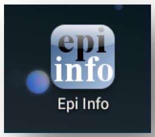
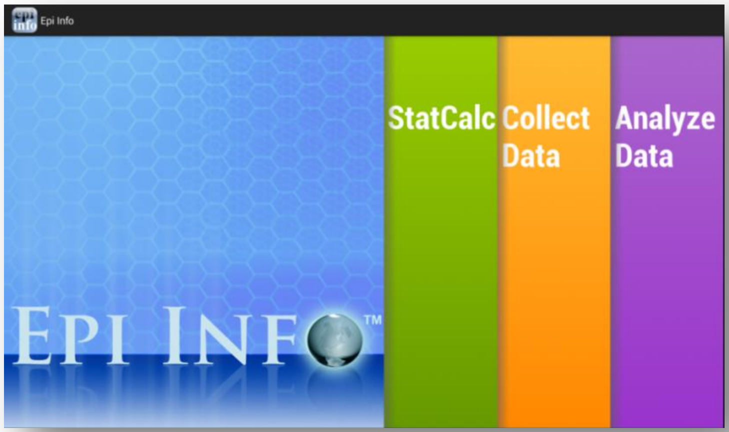
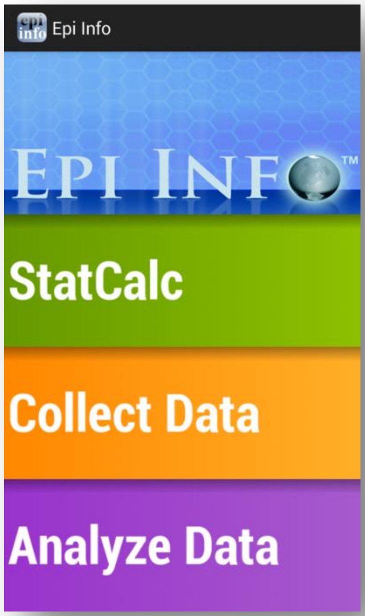


Figure 7.249: Mobile Home screen

1. The Epi Info™ Mobile main menu appears.



Tablet



Phone

Figure 7.250: Epi Info™ Mobile main menu

2. Select **Collect Data**. Click on the triangle on the right hand corner to see a list of available forms. Select the form for data collection from the drop-down list by pressing on the form name.
3. Click **Load**.

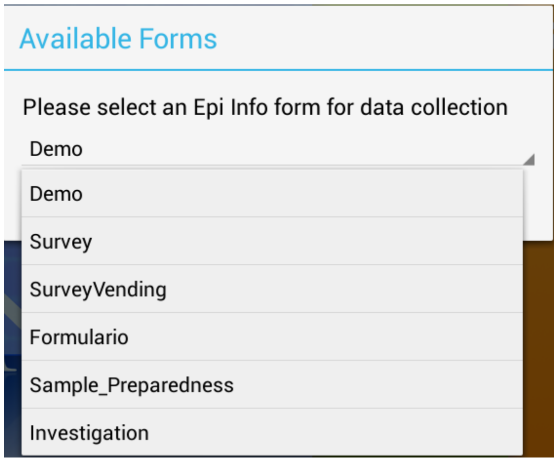
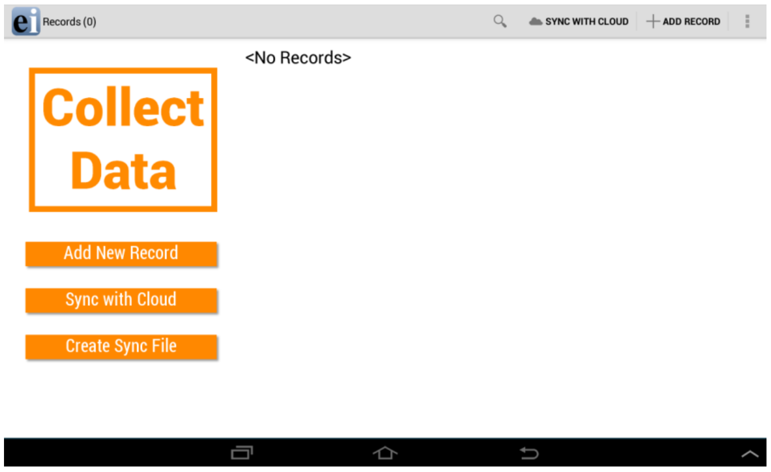


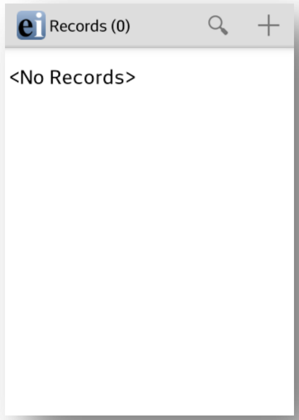
Figure 7.251: Available Forms drop-down list

The Collect Data page appears with a list of all records contained on the mobile device for that form. If there are no records, the record list displays **<No Records>**.

4. To create a new record, select **Add New Record** or **+** from the top right corner of the screen.



Tablet



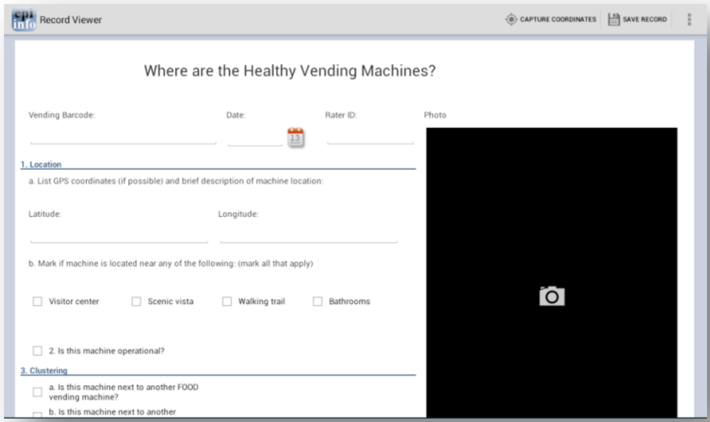
Phone

Figure 7.252: Collect Data main menu

The form appears on the mobile device. Fields appear vertically on a phone while the tablet can display fields horizontally. Users can slide their fingers across the screen to move around the form and tap into the desired field in order to enter data.

Note: Please make sure to check that the date format of the device matches the date format of the PC. Date formats must match between Android device and PC when data is being imported into the central Epi Info 7 database. You can check the date format being used by the device under the Settings>Date and Time>Select Date Format option.

Note: Always navigate using the back button instead of the home button. Use of the home button can cause cache issues that will lead to malfunctions.



Where are the Healthy Vending Machines?

Vending Barcode:

Date:

Rater ID:

Photo

1. Location

a. List GPS coordinates (if possible) and brief description of machine location:

Latitude:

Longitude:

b. Mark if machine is located near any of the following: (mark all that apply)

☐ Visitor center ☐ Scenic vista ☐ Walking trail ☐ Bathrooms

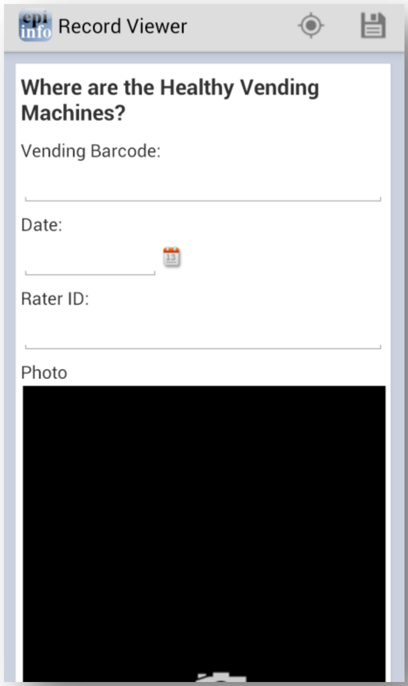
☐ 2. Is this machine operational?

3. Clustering

☐ a. Is this machine next to another FOOD vending machine?

☐ b. Is this machine next to another

Tablet



Where are the Healthy Vending Machines?

Vending Barcode:

Date:

Rater ID:

Photo

Phone

Figure 7.253: Mobile Form View

Text fields

This field allows open text characters. In order to enter data, tap into the field. A keyboard will be displayed to the user in order to enter data. Once data has been entered, click **Done**.

Figure 7.254: Keyboard for Text Fields

Numeric fields

This field allows numerical values. In order to enter data, tap into the field. A keyboard will be displayed to the user allowing only numbers and decimals to be entered. Once data has been entered, click **Done**.

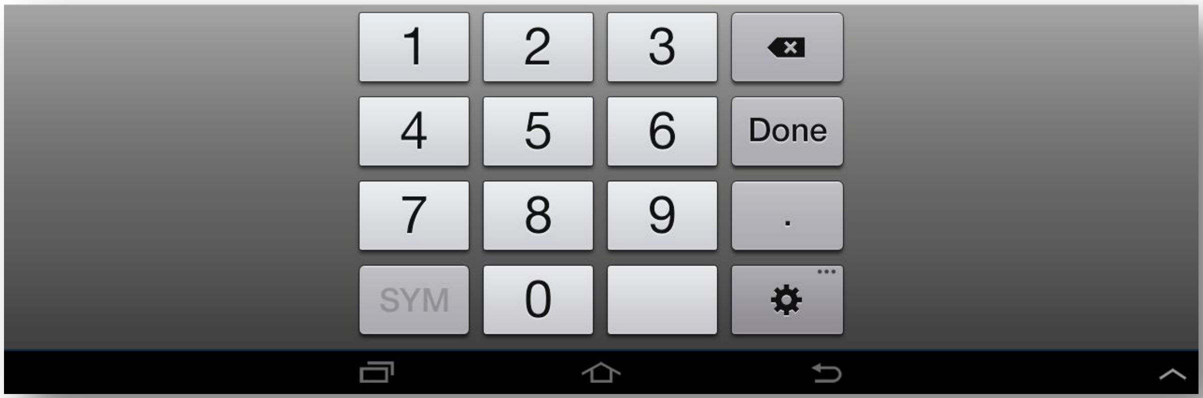
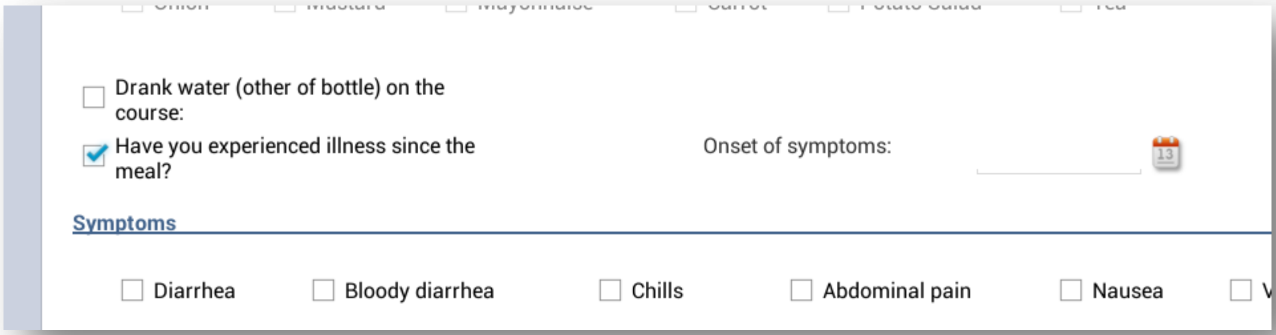


Figure 7.255: Keyboard for Numeric Fields

Date fields

Date fields can be entered by clicking the calendar icon located next to the field. The user can then select a value for each of the data elements (month,day,year). Once data has been entered, click Done.



☐ Drank water (other of bottle) on the course:

☑ Have you experienced illness since the meal?

Onset of symptoms:

Symptoms

☐ Diarrhea ☐ Bloody diarrhea ☐ Chills ☐ Abdominal pain ☐ Nausea

Figure 7.256: Calendar Icon for Dates

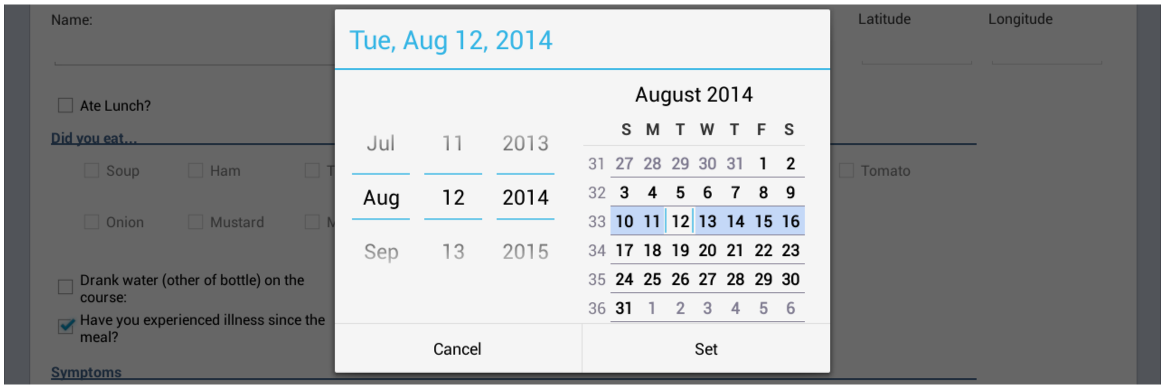


Figure 7.257: Calendar for Selecting Dates

Time fields

Time fields can be entered by clicking the clock icon located next to the field. The user can then specify a value for hour and minutes. Once data has been entered, click Done.

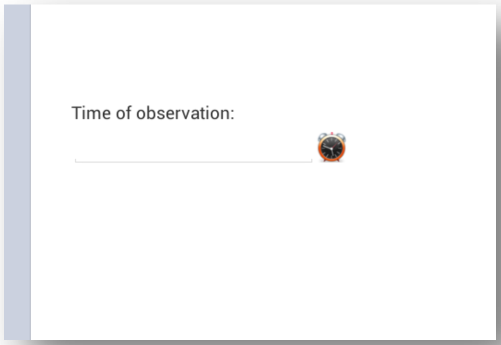


Figure 7.258: Clock Icon

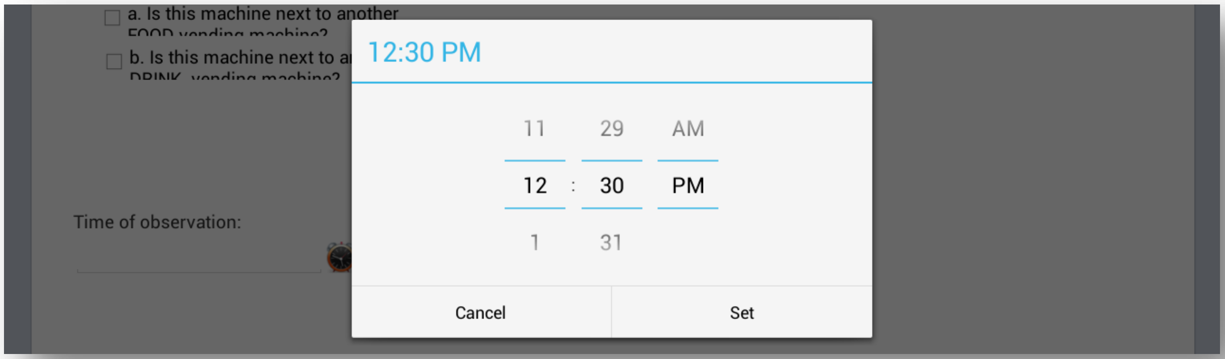
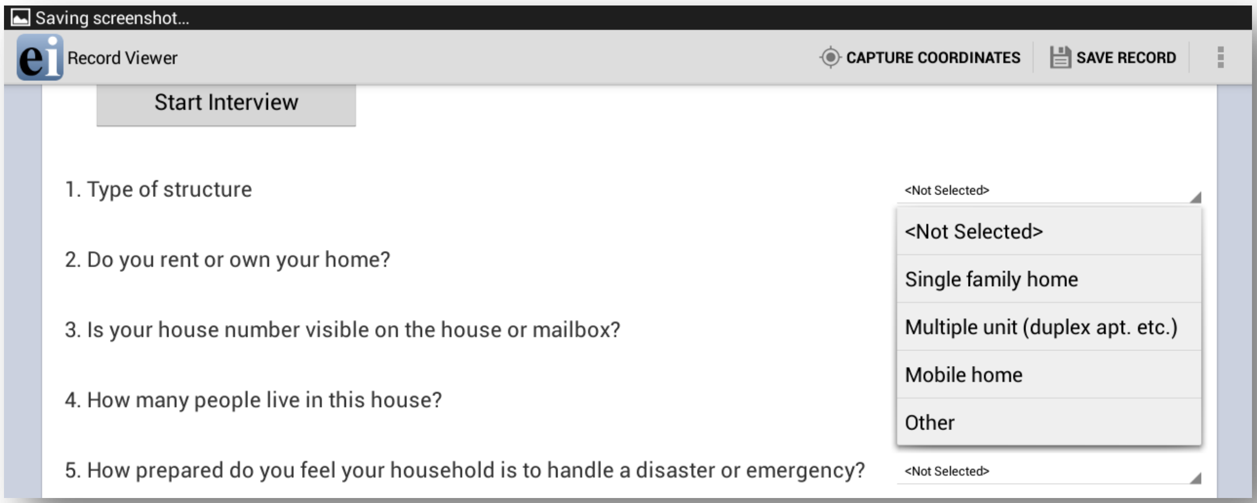

a. Is this machine next to another FOOD vending machine?
b. Is this machine next to a DRINK vending machine?
Time of observation:
12:30 PM
11 29 AM
12 : 30 PM
1 31
Cancel
Set

Figure 7.259: Time Selection Screen

Drop down lists

This field type provides a list of possible values in response to a question therefore limiting the number of options to use. Initially this field will present the user with the prompt <Not Selected>. Tap into the field and select the desired response.



1. Type of structure <Not Selected>

2. Do you rent or own your home?

3. Is your house number visible on the house or mailbox?

4. How many people live in this house?

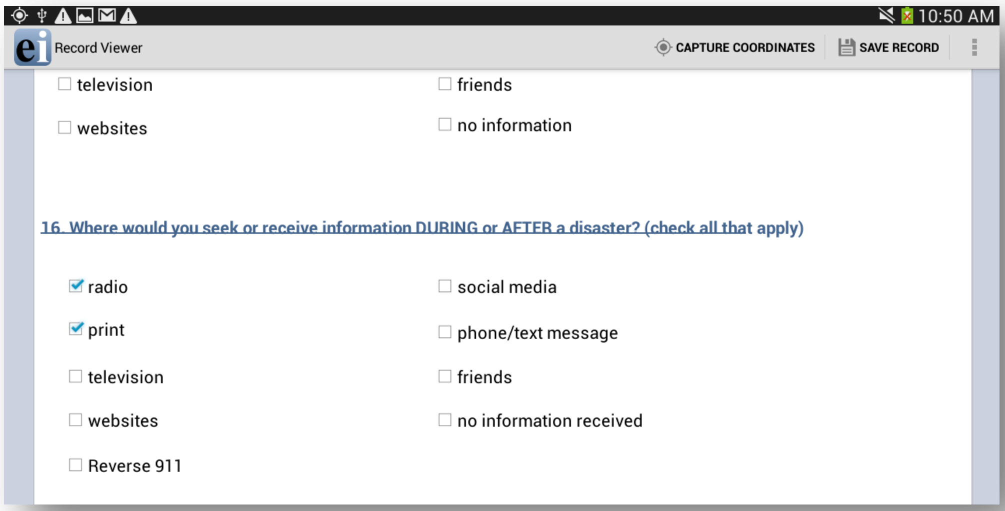
5. How prepared do you feel your household is to handle a disaster or emergency? <Not Selected>

<Not Selected>
Single family home
Multiple unit (duplex apt. etc.)
Mobile home
Other

Figure 7.260: Drop Down Lists

Checkbox fields

The Checkbox field allows you to collect data by checking or unchecking a box. Multiple checkboxes can be used to quickly enter responses that are consistent in the study (e.g. symptoms or foods eaten). Checkbox fields collect binary data such as 0 or 1, True or False, Yes or No. The response is stored in the database as a 1 or 0 where 1 equals Yes and 0 equals No. Checkbox fields do not contain missing values. They are considered to be “No” until they are checked. The following figure provides an example of how Checkbox fields appear on a mobile device.



☐ television ☐ friends
☐ websites ☐ no information

16. Where would you seek or receive information DURING or AFTER a disaster? (check all that apply)

☑ radio ☐ social media
☑ print ☐ phone/text message
☐ television ☐ friends
☐ websites ☐ no information received
☐ Reverse 911

Figure 7.261: Checkbox Fields

Image/Photo Fields

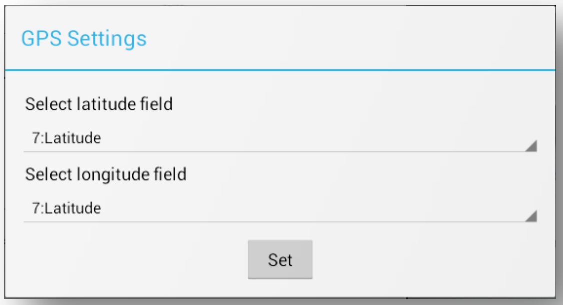
The **Image** field will appear as a **Photo** field on a mobile device. This field allows the user to insert pictures taken with the mobile device onto the form. The photos taken will always stretch to the field shape that appears on the mobile device. The retain image size feature in the Image Field Definition dialog box will not affect the Photo field's functionality. When inserting Image fields, adjust the field size to the anticipated aspect ratio of the mobile device.

1. To add a picture to the form, select the **Photo** field. The mobile device's camera function automatically opens.
2. Take the photo (process varies depending on the mobile device)
3. Select **Save**. Epi Info™ inserts the photo into the Photo field.

Photo fields will appear as the same size regardless of layout on the PC version of Epi Info™ when using a phone.

Capture GPS Coordinates

If the mobile device is connected to the Internet, latitude and longitude coordinates can be captured by selecting **Capture Coordinates**, located in the top right corner of the screen. Select the field containing the latitude coordinates from the **Select latitude field** drop-down list. Select the field containing the longitude coordinates from the **Select longitude field** drop-down list. Select **Set**.


GPS Settings

Select latitude field

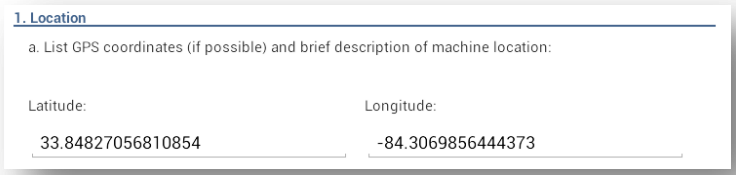
7:Latitude

Select longitude field

7:Latitude

Figure 7.262: GPS Settings

The coordinates appear in the latitude and longitude fields.


1. Location

a. List GPS coordinates (if possible) and brief description of machine location:

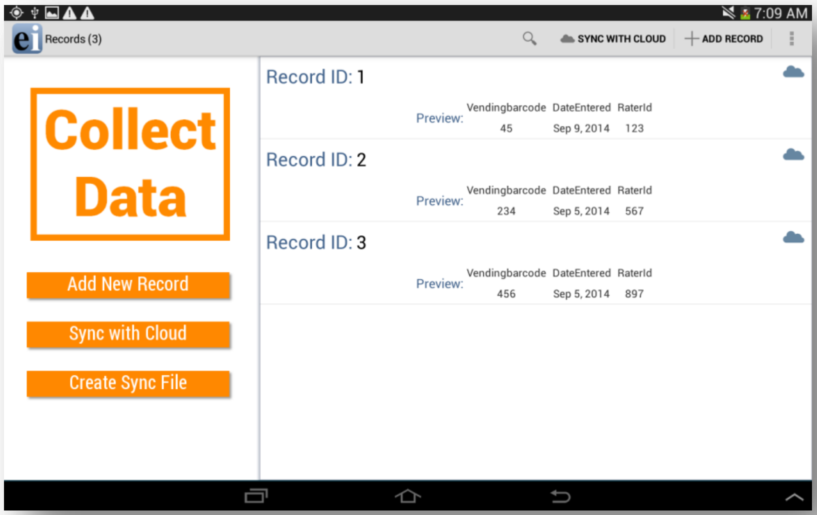
Latitude: 33.84827056810854

Longitude: -84.3069856444373

Figure 7.263: Latitude and Longitude Coordinates

Save Record

To save the form, select Save Record in the top right corner of the screen. An incomplete form can be saved and finished at a later time. The data is stored with a Record ID and Preview information.



Record ID: 1

Preview:	Vendingbarcode	DateEntered	RaterId
	45	Sep 9, 2014	123

Record ID: 2

Preview:	Vendingbarcode	DateEntered	RaterId
	234	Sep 5, 2014	567

Record ID: 3

Preview:	Vendingbarcode	DateEntered	RaterId
	456	Sep 5, 2014	897

Figure 7.264: Record List

Edit Record
Select the record you wish to edit from the Record List. The form appears on the device and is ready for editing. At any point, select the **Save Record** button to save the edited record.

Delete Record
Hold down on the desired Record from the Record List and a menu will appear. Select **Delete**.

Search Record
To search the Record list, select **Search** in the top right corner of the Collect Data menu. The Epi Info Record list (top left corner of the screen) becomes blank with a blinking cursor. The Search function allows records to be located based on a series of matched variables. Type in the search criteria and press the search button. The records that match the search criteria appear in the record list. When searching for a record, you will need to specify the fieldname, the = operator, and then the value. If you are searching for a text field, use single quotes around the value being searched. For example **PATIENTNAME='Walter'**if you are searching for "Walter" as the patient name in your tablet.

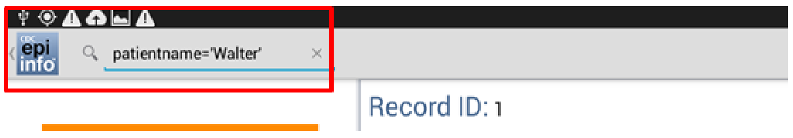
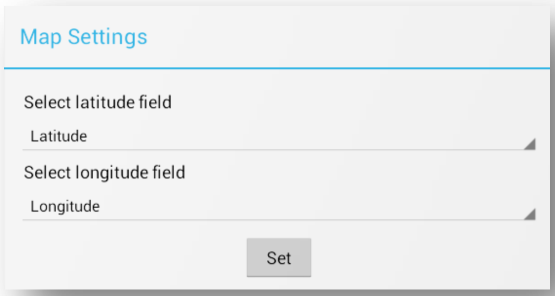


Figure 7.265: Syntax for Search text field

You can use the AND or the OR operator to use more than one variable in your search criteria. If you are searching for a date or numeric field, simply type the value to be searched (i.e. DOB=06/12/1999 or AGE=35) without using any quotes.

Maps
The Maps feature allows users to geographically display case data on a map. The mobile version contains the ability to plot GPS coordinates on a Google Map base map. Internet connectivity is required to display maps on your mobile device using Google Maps. For additional mapping functionality, transfer the data to a computer and use the Epi Info™ Maps tool (See Maps).

1. Select **Generate Map** from the **Collect Data** menu or the options menu. Generate Map is only available through the options menu on a phone. A **Map Settings** dialog box appears.
2. Select the field containing the latitude coordinates from the **Select latitude field** drop-down list.
3. Select the field containing the longitude coordinates from the **Select longitude field** drop-down list.
4. Select **Set**.


Map Settings

Select latitude field: Latitude

Select longitude field: Longitude

Figure 7.266: Maps Settings

The map appears displaying the location in Google Maps.

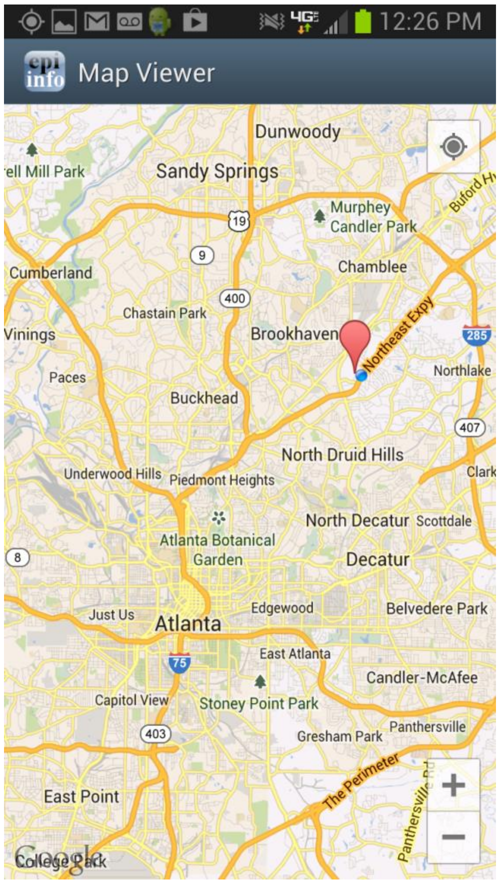
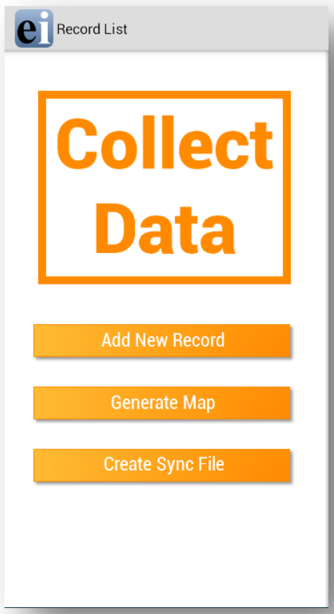
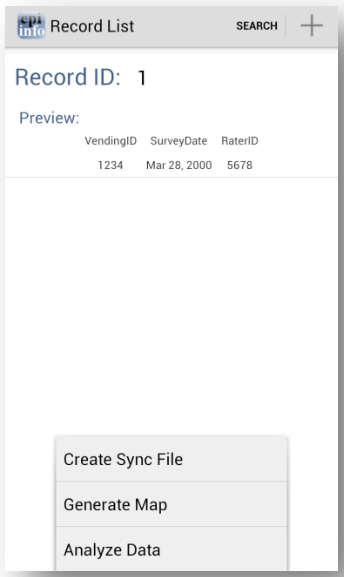


Figure 7.267: Maps

Transfer Data from Mobile Device

1. Select **Create Sync File** from the Collect Data menu or from the options menu. This feature is only accessible from the options menu on a phone.



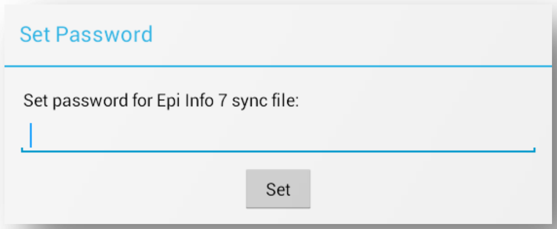


Phone

Tablet

Figure 7.268: Create Sync File

2. Create a **password** for the sync file. This password will be required to access the sync file and download the information to a computer. There are no password requirements; however, it is recommended that users create passwords to ensure secure information transfer with encrypted files.

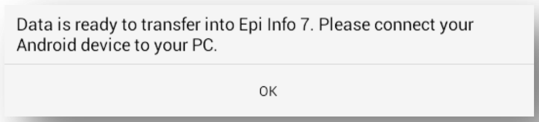

Set Password

Set password for Epi Info 7 sync file:

Set

Figure 7.269: Set Sync File Password

3. Click **OK** and connect the device to your PC.


Data is ready to transfer into Epi Info 7. Please connect your Android device to your PC.

OK

Figure 7.270: Sync File Created

4. Open the Epi Info™ software on your PC.
5. From the Epi Info™ main menu, select **Enter Data** or select **Tools > Enter Data**. The Enter tool opens.
6. Open the project and form used to collect the data on the mobile device.
7. From the File menu, select **Import Data > From mobile device**.

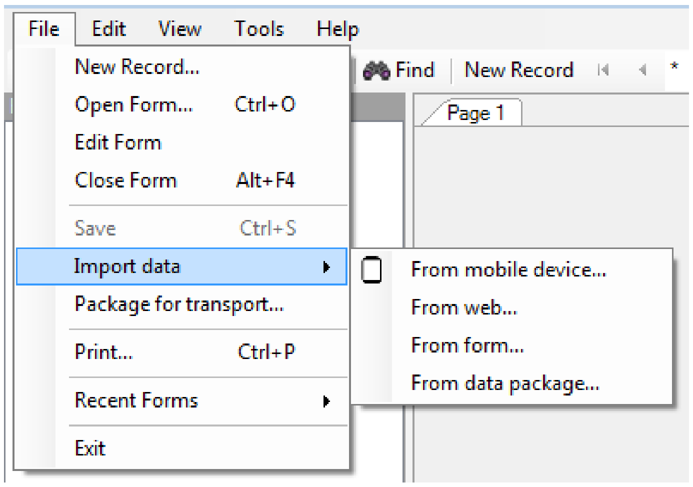
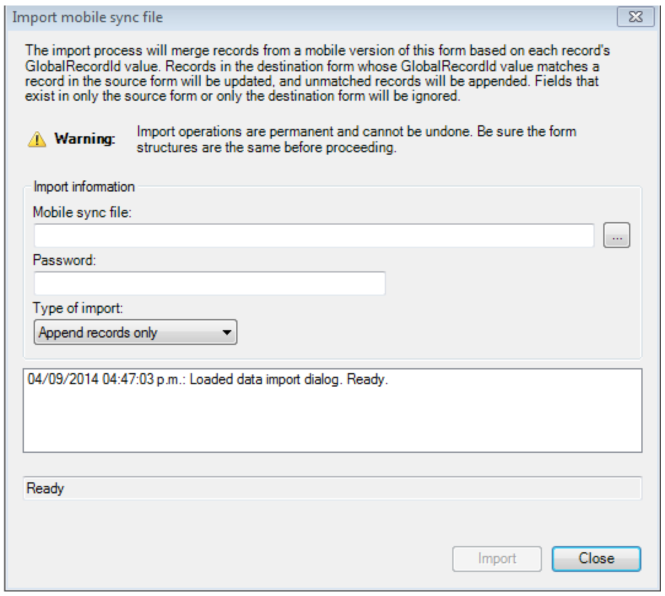


Figure 7.271: Import Data from mobile device

The **Import phone data** dialog box appears.


Import mobile sync file

The import process will merge records from a mobile version of this form based on each record's GlobalRecordId value. Records in the destination form whose GlobalRecordId value matches a record in the source form will be updated, and unmatched records will be appended. Fields that exist in only the source form or only the destination form will be ignored.

Warning: Import operations are permanent and cannot be undone. Be sure the form structures are the same before proceeding.

Import information

Mobile sync file:

Password:

Type of import: Append records only

04/09/2014 04:47:03 p.m.: Loaded data import dialog. Ready.

Ready

Figure 7.272: Import Mobile Sync File Dialog Box

8. Click the **ellipses** button next to the **Android sync file** text box and select the sync file created with the mobile device. The file will be located in the **Downloads > EpiInfo> SyncFiles** folder on the mobile device.
9. Click **Open**. Note the file format, "Epi Info for Android" sync files.

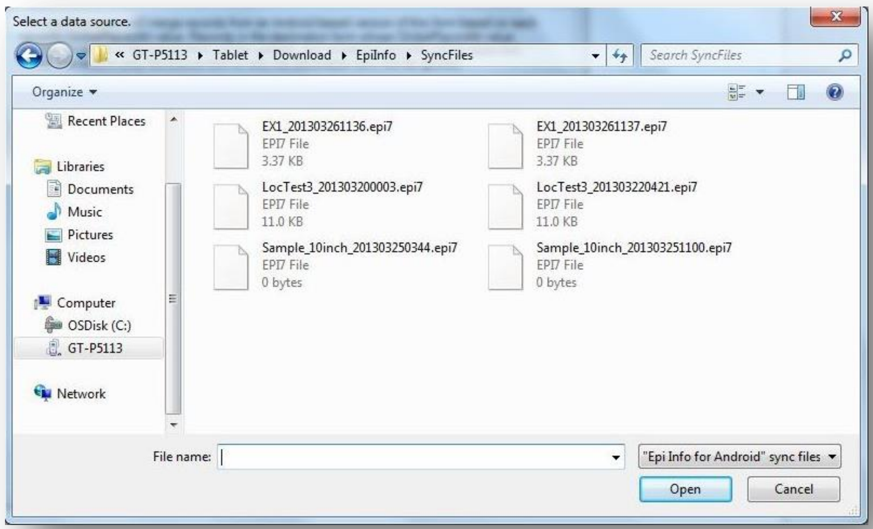


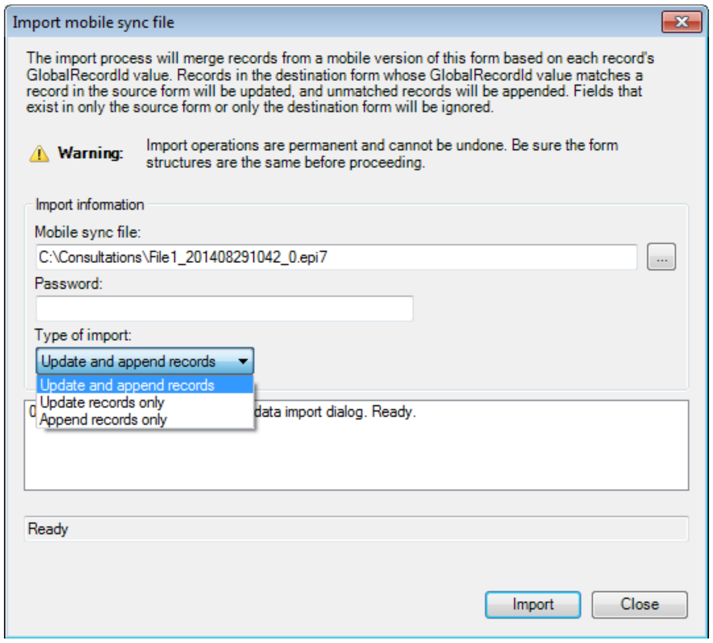
Figure 7.273: Select a Data Source Dialog Box

10. Enter the password created for the sync file in the **Password** field.
11. Select the **Type of Import** from the drop-down list.

- **Update and append records** – Records in the destination form containing the same GlobalRecordId values as the source form will be updated (overwritten) if there is new information. However, values in the destination table will never be overwritten with a null or missing value. All other records will be appended (added) to the end of the table.

- **Update records only** – Records in the destination form containing the same GlobalRecordId values as the source form will be updated (overwritten) if there is new information. However, values in the destination table will never be overwritten with a null or missing value. All other records will be ignored.

- **Append records only** – All records in the source form will be appended (added) to the end of the table. No records will be updated (overwritten).



The import process will merge records from a mobile version of this form based on each record's GlobalRecordId value. Records in the destination form whose GlobalRecordId value matches a record in the source form will be updated, and unmatched records will be appended. Fields that exist in only the source form or only the destination form will be ignored.

Warning: Import operations are permanent and cannot be undone. Be sure the form structures are the same before proceeding.

Import information

Mobile sync file: C:\Consultations\File1_201408291042_0.epi7

Password:

Type of import: Update and append records

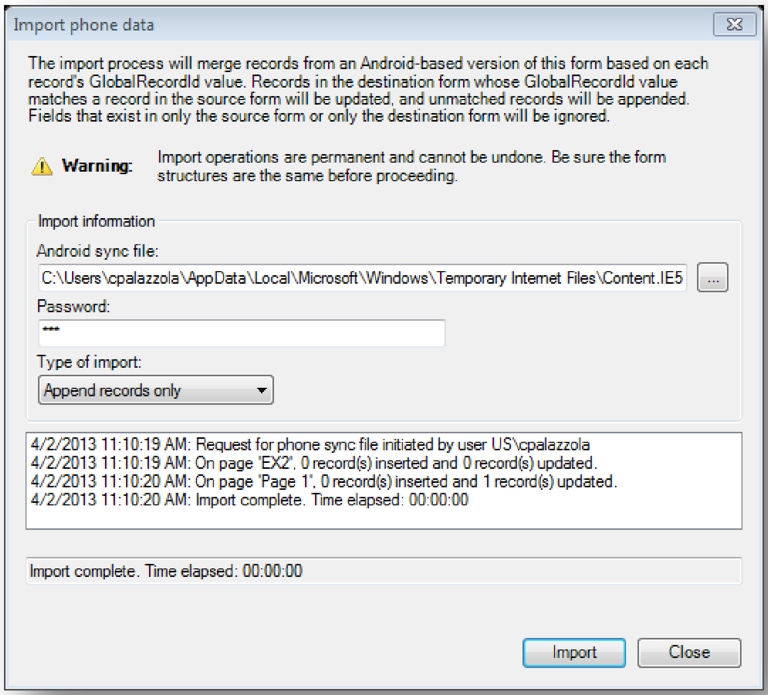
data import dialog. Ready.

Ready

Figure 7.274: Type of Import Drop-Down List

Note: Import operations are permanent and irreversible. Be sure that the form structures are the same (including all form names and form table names) before proceeding.

12. Click **Import**. *Import Complete* will appear at the bottom of the dialog box upon successful completion of the import.



Import phone data

The import process will merge records from an Android-based version of this form based on each record's GlobalRecordId value. Records in the destination form whose GlobalRecordId value matches a record in the source form will be updated, and unmatched records will be appended. Fields that exist in only the source form or only the destination form will be ignored.

Warning: Import operations are permanent and cannot be undone. Be sure the form structures are the same before proceeding.

Import information

Android sync file: C:\Users\cpalazzola\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5

Password: ***

Type of import: Append records only

```
4/2/2013 11:10:19 AM: Request for phone sync file initiated by user US\cpalazzola
4/2/2013 11:10:19 AM: On page 'EX2', 0 record(s) inserted and 0 record(s) updated.
4/2/2013 11:10:20 AM: On page 'Page 1', 0 record(s) inserted and 1 record(s) updated.
4/2/2013 11:10:20 AM: Import complete. Time elapsed: 00:00:00
```

Import complete. Time elapsed: 00:00:00

Figure 7.275: Import Complete

13. Click **Close**.

The data displays in the Enter tool at the end of the Record List. This information is now ready to be analyzed by Classic Analysis or Visual Dashboard.

NOTE: You might need to disconnect the cable from the Android device into your PC and reconnect it in order for the PC to recognize updated files on the device. In some devices, users had to restart the device in order for the PC to recognize updated files on the device.

Configuring cloud synchronization with Windows Azure

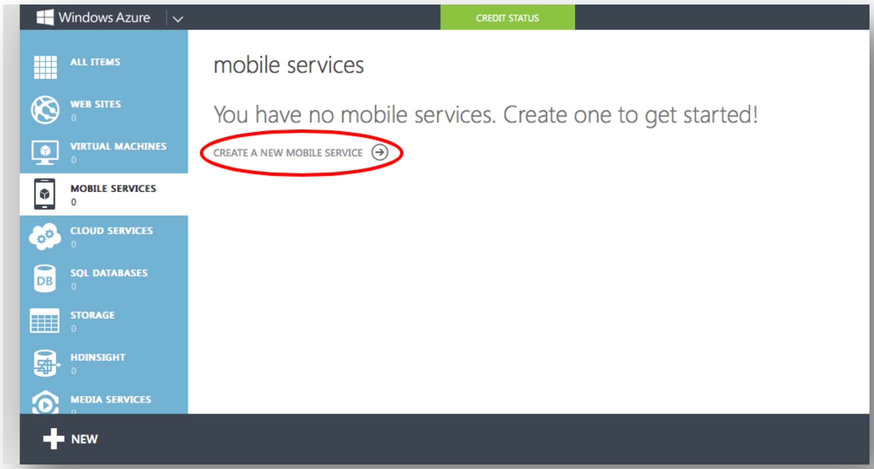
The companion for Android application allows you to synchronize with a Windows Azure cloud service and consume the data using Epi Info™ Companion for Android to allow a team of epidemiologists to more efficiently collect data together. Data can be collected offline and synchronized with team members when Internet connectivity becomes available. If you are always connected, data shows up in the cloud instantly to allow for real-time data analysis and situational awareness.

You will need to first create a Windows Azure account at [Windows Azure Home](http://windowsazure.com/) (http://windowsazure.com/). The free tier is sufficient for many data collection activities. **Please note that you MUST get organizational approval first if you are using this service for official purposes!**

Please complete the steps documented below in order to set up your Companion for Android application with Windows Azure

Add a Mobile service

1. Select the *Create a New Mobile Service* option

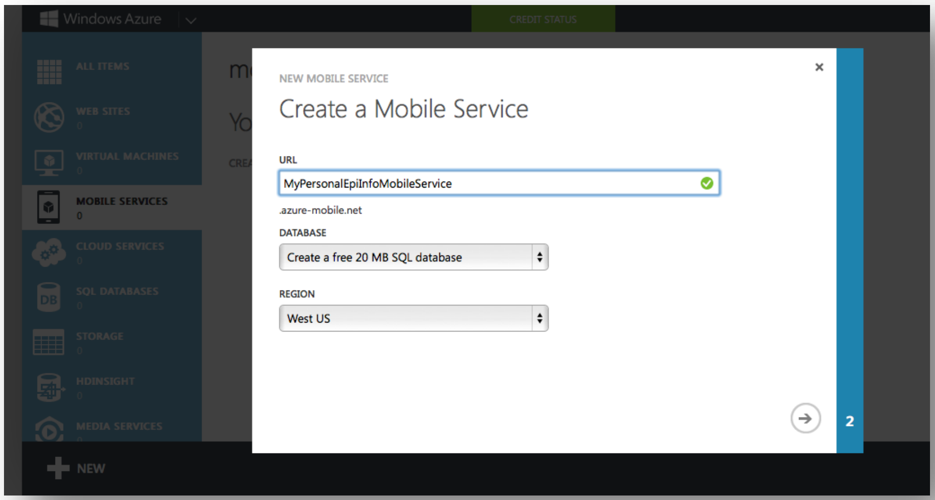


mobile services

You have no mobile services. Create one to get started!

Figure 7.276: Create a New Mobile Service

2. Provide a unique Mobile Service Name. You will need to either remember *(or copy)* the Mobile Service Name and provide it in the configuration settings of the application.



NEW MOBILE SERVICE

Create a Mobile Service

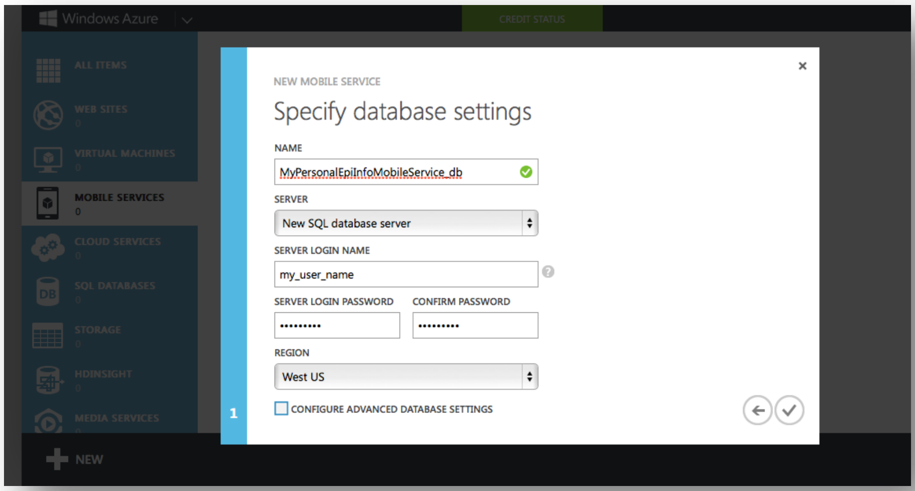
URL: MyPersonalEpiInfoMobileService .azure-mobile.net

DATABASE: Create a free 20 MB SQL database

REGION: West US

Figure 7.277: Mobile Service Name

3. Create an SQL Server database. Specify a database name, Login Name, and Password.



NEW MOBILE SERVICE

Specify database settings

NAME: MyPersonalEpiInfoMobileService_db

SERVER: New SQL database server

SERVER LOGIN NAME: my_user_name

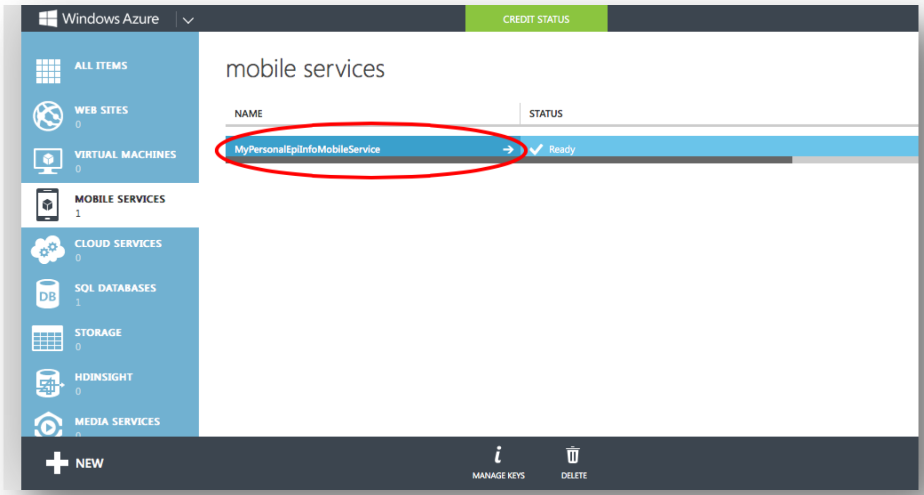
SERVER LOGIN PASSWORD: ••••••••• CONFIRM PASSWORD: •••••••••

REGION: West US

CONFIGURE ADVANCED DATABASE SETTINGS

Figure 7.278: Database Settings

4. Once completed, click on the Mobile Service Name

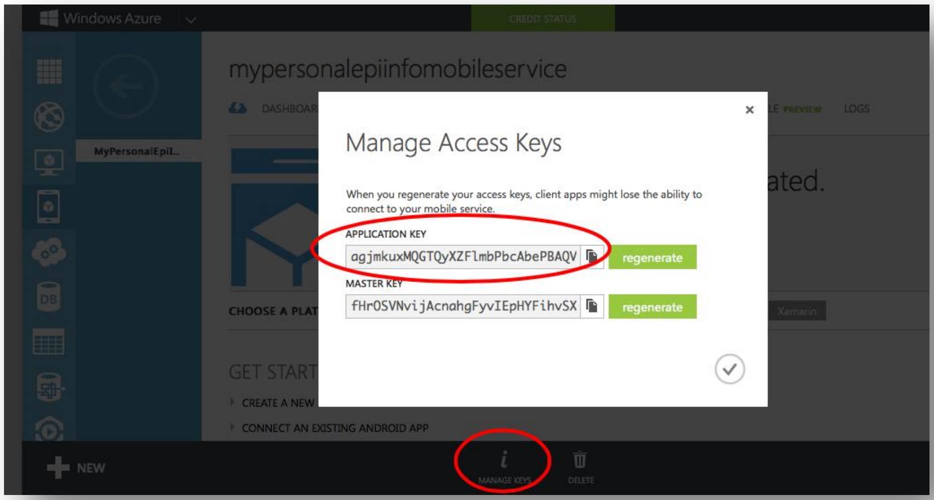


mobile services

NAME	STATUS
MyPersonalEpiInfoMobileService	Ready

Figure 7.279: Accessing Mobile Service Configuration

5. Click on the **Manage Access Keys** link. You will need to copy this Application Key into the Companion for Android configuration settings in order to authenticate against your service.



Manage Access Keys

When you regenerate your access keys, client apps might lose the ability to connect to your mobile service.

APPLICATION KEY
agjmkuxMQGTQyXZFlmbPbcAbePBAQV
regenerate

MASTER KEY
fHrOSVNvijAcnahgFyvIEpHYFihvSX
regenerate

Figure 7.280: Managing Access Keys link

Configuring the application to use the settings

1. From the Companion for Android main menu, click on the Settings option.

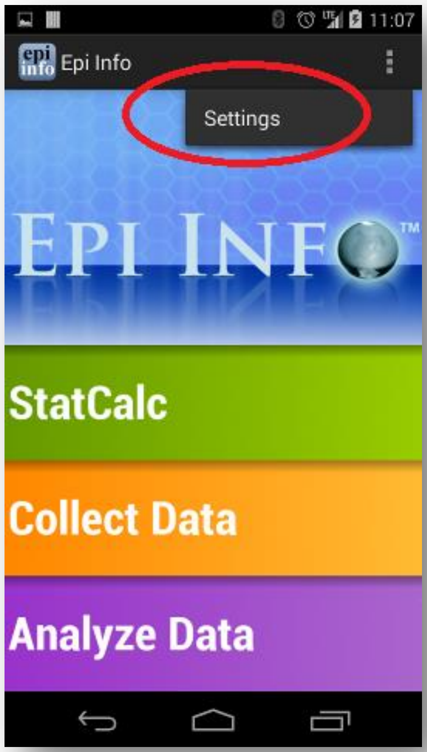
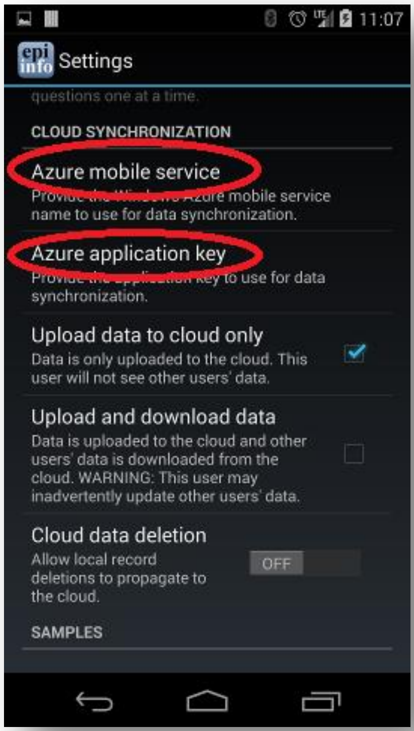


Figure 7.281: Companion for Android Configuration Settings

2. Provide the Windows Azure mobile service name to use for data synchronization in the Azure mobile service section. Also, provide the Windows Azure application key to use for data synchronization in the Azure application key section.


11:07

Settings

questions one at a time.

CLOUD SYNCHRONIZATION

Azure mobile service
Provide the Windows Azure mobile service name to use for data synchronization.

Azure application key
Provide the application key to use for data synchronization.

Upload data to cloud only ✓
Data is only uploaded to the cloud. This user will not see other users' data.

Upload and download data ☐
Data is uploaded to the cloud and other users' data is downloaded from the cloud. WARNING: This user may inadvertently update other users' data.

Cloud data deletion OFF
Allow local record deletions to propagate to the cloud.

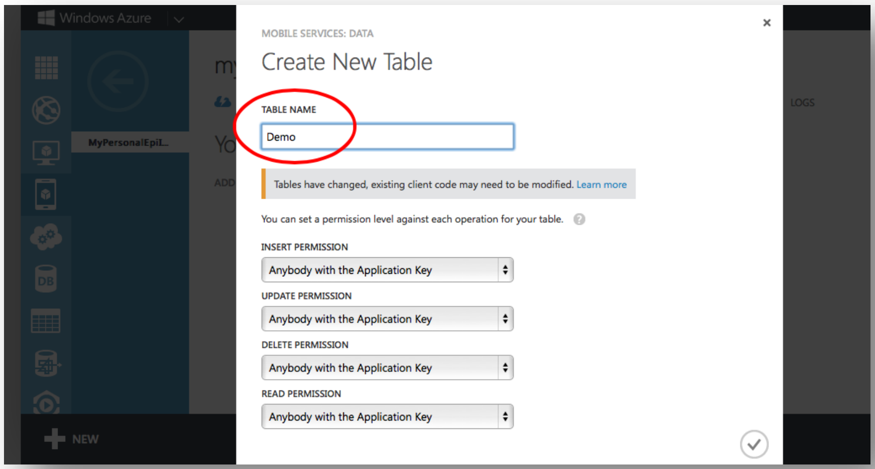
SAMPLES

Figure 7.282: Specify mobile service name

Receiving Data and Creating a Data Table in Windows Azure

The last step required in order to synchronize data between the device and the cloud would consist of creating a data table in the cloud. In order to do so, please complete the following steps:

1. Login into your Windows Azure account.
2. Click on *Mobile Services*.
3. Click on the corresponding mobile service.
4. Click on *Data*.
5. Click on *Create* (located at the bottom of the screen)
6. Provide a table name. Please Note that this Table Name must **exactly match the corresponding Form Name in the Companion for Android application.**


MOBILE SERVICES: DATA

Create New Table

TABLE NAME

Demo

Tables have changed, existing client code may need to be modified. Learn more

You can set a permission level against each operation for your table.

INSERT PERMISSION

Anybody with the Application Key

UPDATE PERMISSION

Anybody with the Application Key

DELETE PERMISSION

Anybody with the Application Key

READ PERMISSION

Anybody with the Application Key

Figure 7.283: Create data table in cloud service

7. Click on the check mark icon to complete the table creation process.

At this point, you should be able to start entering data into your android device and synchronizing data using the Sync with Cloud option of the Collect data module.

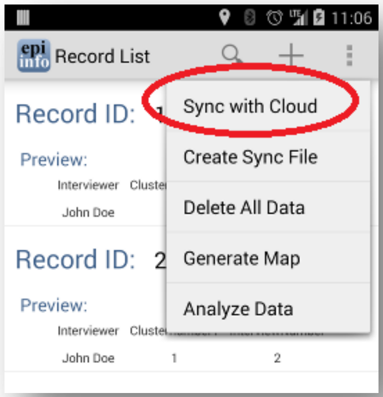
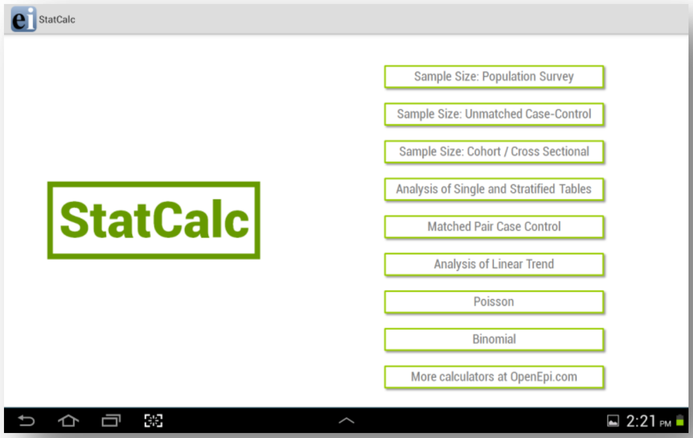


Figure 7.284: Synchronize data with cloud

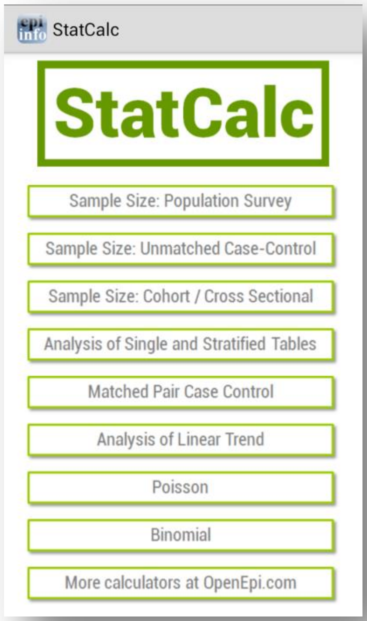
StatCalc Epidemiologic Calculators

The StatCalc tool is a calculator capable of analysis pre and post data collection. The version available on the Companion for Android has similar functionality to the full Epi

Info™ StatCalc tool and is capable of performing the analyses listed in the StatCalc menu listed below.


StatCalc
StatCalc
Sample Size: Population Survey
Sample Size: Unmatched Case-Control
Sample Size: Cohort / Cross Sectional
Analysis of Single and Stratified Tables
Matched Pair Case Control
Analysis of Linear Trend
Poisson
Binomial
More calculators at OpenEpi.com
2:21 PM

Tablet


StatCalc
StatCalc
Sample Size: Population Survey
Sample Size: Unmatched Case-Control
Sample Size: Cohort / Cross Sectional
Analysis of Single and Stratified Tables
Matched Pair Case Control
Analysis of Linear Trend
Poisson
Binomial
More calculators at OpenEpi.com

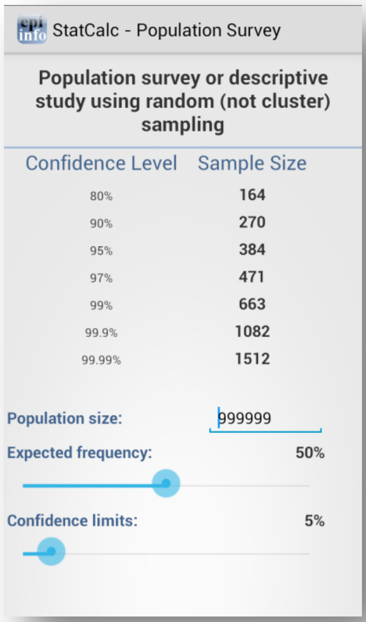
Phone

Figure 7.285: StatCalc main menu

For additional information regarding StatCalc, reference the StatCalc section of the user guide.

Sample Size: Population Survey

The population survey calculates how many samples are recommended for a survey given a population size, expected frequency, design effect, the number of clusters and the desired confidence level.



StatCalc - Population Survey

Population survey or descriptive study using random (not cluster) sampling

Confidence Level	Sample Size
80%	164
90%	270
95%	384
97%	471
99%	663
99.9%	1082
99.99%	1512

Population size: 999999

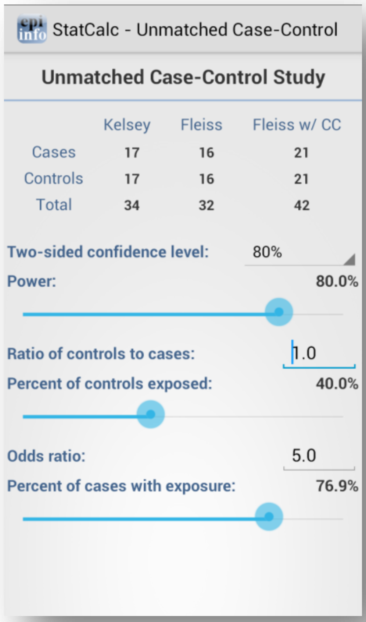
Expected frequency: 50%

Confidence limits: 5%

Figure 7.286: Population Survey

Sample Size: Unmatched Case-Control

The Unmatched Case Control study calculates how many samples are recommended for a study given the power, proportion of unexposed vs. exposed, percentage outcome in exposed group, percentage outcome in the unexposed group, and the desired confidence level.



StatCalc - Unmatched Case-Control

Unmatched Case-Control Study

	Kelsey	Fleiss	Fleiss w/ CC
Cases	17	16	21
Controls	17	16	21
Total	34	32	42

Two-sided confidence level: 80%

Power: 80.0%

Ratio of controls to cases: 1.0

Percent of controls exposed: 40.0%

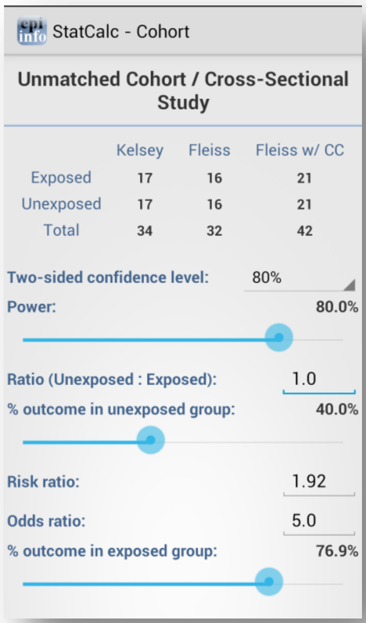
Odds ratio: 5.0

Percent of cases with exposure: 76.9%

Figure 7.287: Unmatched Case-Control

Sample Size: Cohort / Cross Sectional

The Unmatched Case Control study calculates how many samples are recommended for a study given the power, proportion of unexposed vs. exposed, percentage outcome in exposed group, percentage outcome in the unexposed group, and the desired confidence level.



StatCalc - Cohort

Unmatched Cohort / Cross-Sectional Study

	Kelsey	Fleiss	Fleiss w/ CC
Exposed	17	16	21
Unexposed	17	16	21
Total	34	32	42

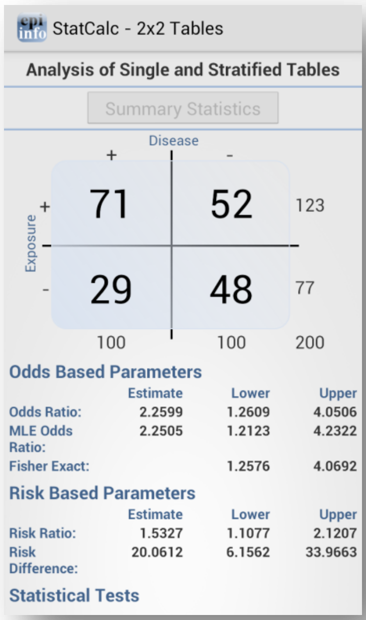
Two-sided confidence level: 80%
Power: 80.0%
Ratio (Unexposed : Exposed): 1.0
% outcome in unexposed group: 40.0%
Risk ratio: 1.92
Odds ratio: 5.0
% outcome in exposed group: 76.9%

Figure 7.288: Cohort/Cross Sectional

Analysis of Single and Stratified Tables

2 x 2 tables are frequently used in epidemiology to explore associations between exposures to risk factors and disease or other outcomes. The table in StatCalc has Exposure on the left and Outcome across the top. Given a yes-no or other two-choice response describing disease and another describing exposure to a risk factor, StatCalc produces several kinds of statistics that test for relationships between exposure and disease.

Stratifying a dataset separates the population into distinct categories based on a parameter (i.e. sex). If confounding is present, associations between disease and exposure can be missed or falsely detected.



	+	-	
+	71	52	123
-	29	48	77
	100	100	200

Odds Based Parameters

	Estimate	Lower	Upper
Odds Ratio:	2.2599	1.2609	4.0506
MLE Odds Ratio:	2.2505	1.2123	4.2322
Fisher Exact:		1.2576	4.0692

Risk Based Parameters

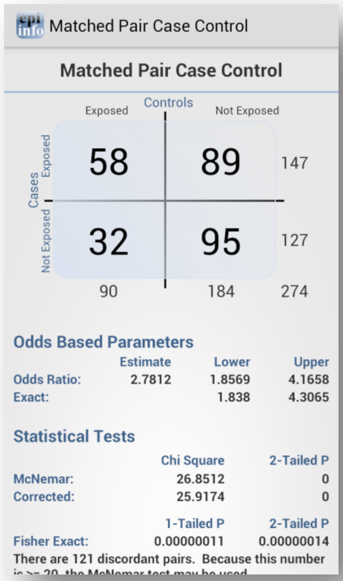
	Estimate	Lower	Upper
Risk Ratio:	1.5327	1.1077	2.1207
Risk Difference:	20.0612	6.1562	33.9663

Statistical Tests

Figure 7.289: Single and Stratified Tables

Matched Pair Case Control

The Matched Pair Case-Control Study calculates the statistical relationship between exposures and the likelihood of becoming ill in a given patient population. This study is used to investigate a cause of an illness by selecting a non-ill person as the control and matching the control to a case. The control can be matched to one or more criteria.


Matched Pair Case Control

Matched Pair Case Control

		Controls Exposed	Controls Not Exposed	
Cases	Exposed	58	89	147
	Not Exposed	32	95	127
		90	184	274

Odds Based Parameters

	Estimate	Lower	Upper
Odds Ratio:	2.7812	1.8569	4.1658
Exact:		1.838	4.3065

Statistical Tests

	Chi Square	2-Tailed P
McNemar:	26.8512	0
Corrected:	25.9174	0

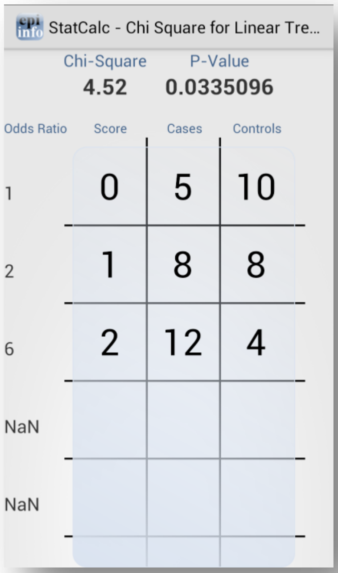
	1-Tailed P	2-Tailed P
Fisher Exact:	0.00000011	0.00000014

There are 121 discordant pairs. Because this number

Figure 7.290: Match Pair Case Control

Analysis of Linear Trend

The Linear Trend function calculates the odds ratio, chi square for linear trend, and p-value statistics based on the response to an exposure score and whether or not the patient has become ill. The exposure score is a measured outcome from a study that states the level of exposure the patient received.


StatCalc - Chi Square for Linear Tre...

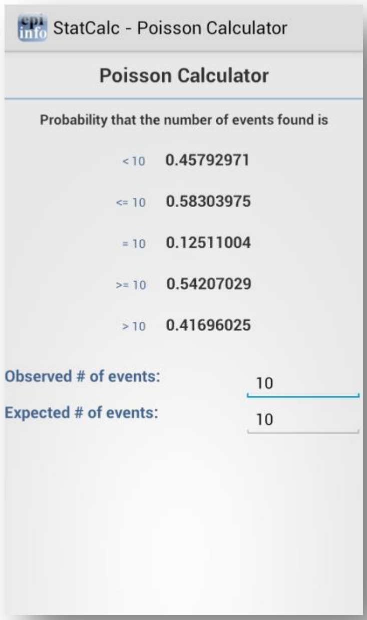
Chi-Square	P-Value
4.52	0.0335096

Odds Ratio	Score	Cases	Controls
1	0	5	10
2	1	8	8
6	2	12	4
NaN			
NaN			

Figure 7.291: Analysis of Linear Trend

Poisson

The Poisson distribution states the probability that a number of positive outcomes occurs based on the expected number of positive outcomes. To analyze the Poisson distribution, enter the expected number of positive outcomes in Expected # of events and the value of positive outcomes you would like to determine the probability of in Observed # of events.



StatCalc - Poisson Calculator

Poisson Calculator

Probability that the number of events found is

< 10	0.45792971
<= 10	0.58303975
= 10	0.12511004
>= 10	0.54207029
> 10	0.41696025

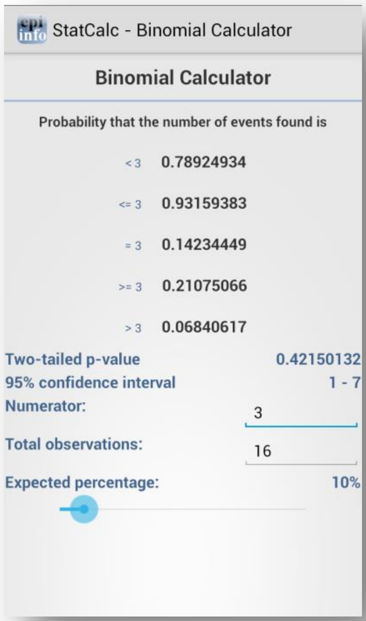
Observed # of events: 10

Expected # of events: 10

Figure 7.292: Poisson Distribution

Binomial

The binomial distribution states the probability that a number of positive outcomes occurs given the expected percentage of positive outcomes and the total number of observations taken.


StatCalc - Binomial Calculator

Binomial Calculator

Probability that the number of events found is

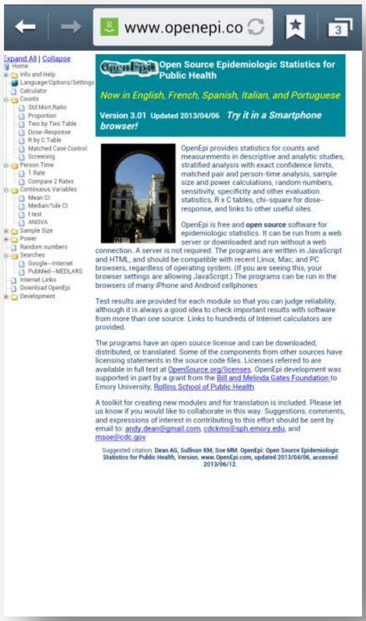
< 3	0.78924934
<= 3	0.93159383
= 3	0.14234449
>= 3	0.21075066
> 3	0.06840617

Two-tailed p-value	0.42150132
95% confidence interval	1 - 7
Numerator:	3
Total observations:	16
Expected percentage:	10%

Figure 7.293: Binomial Distribution

OpenEpi.com

This website is an open source web tool that provides additional epidemiologic statistics.



Open Source Epidemiologic Statistics for Public Health

Now in English, French, Spanish, Italian, and Portuguese

Version 3.01 Updated 2013/04/06 ***Try it in a Smartphone browser!***

OpenEpi provides statistics for counts and measurements in descriptive and analytic studies, stratified analysis with exact confidence limits, matched pair and person-time analysis, sample size and power calculations, random numbers, sensitivity, specificity and other evaluation statistics, R x C tables, chi-square for dose-response, and links to other useful sites.

OpenEpi is free and **open source** software for epidemiologic statistics. It can be run from a web server or downloaded and run without a web connection. A server is not required. The programs are written in JavaScript and HTML, and should be compatible with recent Linux, Mac, and PC browsers, regardless of operating system. (If you are seeing this, your browser settings are allowing JavaScript.) The programs can be run in the browsers of many iPhone and Android cellphones

Test results are provided for each module so that you can judge reliability, although it is always a good idea to check important results with software from more than one source. Links to hundreds of Internet calculators are provided.

The programs have an open source license and can be downloaded, distributed, or translated. Some of the components from other sources have licensing statements in the source code files. Licenses referred to are available in full text at OpenSource.org/licenses. OpenEpi development was supported in part by a grant from the Bill and Melinda Gates Foundation to Emory University, Rollins School of Public Health.

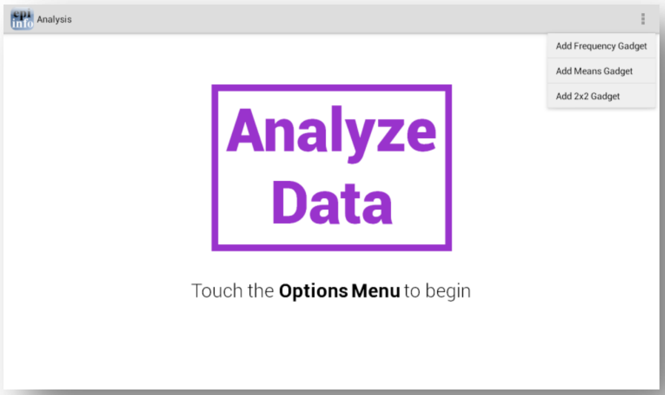
A toolkit for creating new modules and for translation is included. Please let us know if you would like to collaborate in this way. Suggestions, comments, and expressions of interest in contributing to this effort should be sent by email to: andy.dean@gmail.com, cdckms@sph.emory.edu, and msoe@cdc.gov

Suggested citation: **Dean AG, Sullivan KM, Soe MM. OpenEpi: Open Source Epidemiologic Statistics for Public Health, Version. www.OpenEpi.com, updated 2013/04/06, accessed 2013/06/12.**

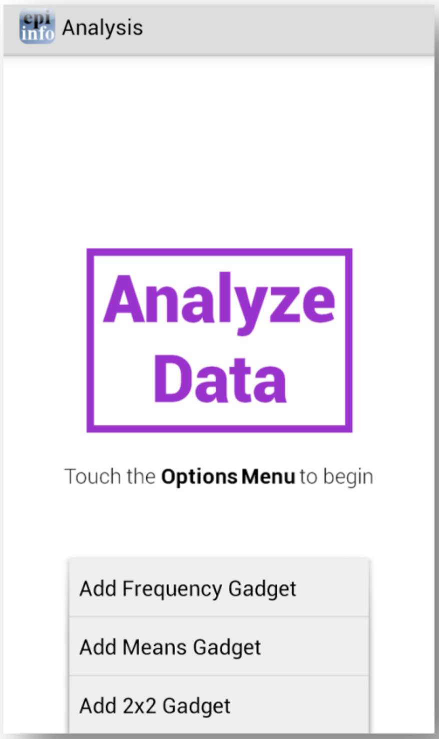
Figure 7.294: OpenEpi.com

Analyze Data

The Analyze Data tool contains three commonly used analysis functions from the Visual Dashboard tool. For additional functionality, transfer the data to a PC and use the Epi Info™ Visual Dashboard tool (refer to Visual Dashboard section of the user guide).



Tablet



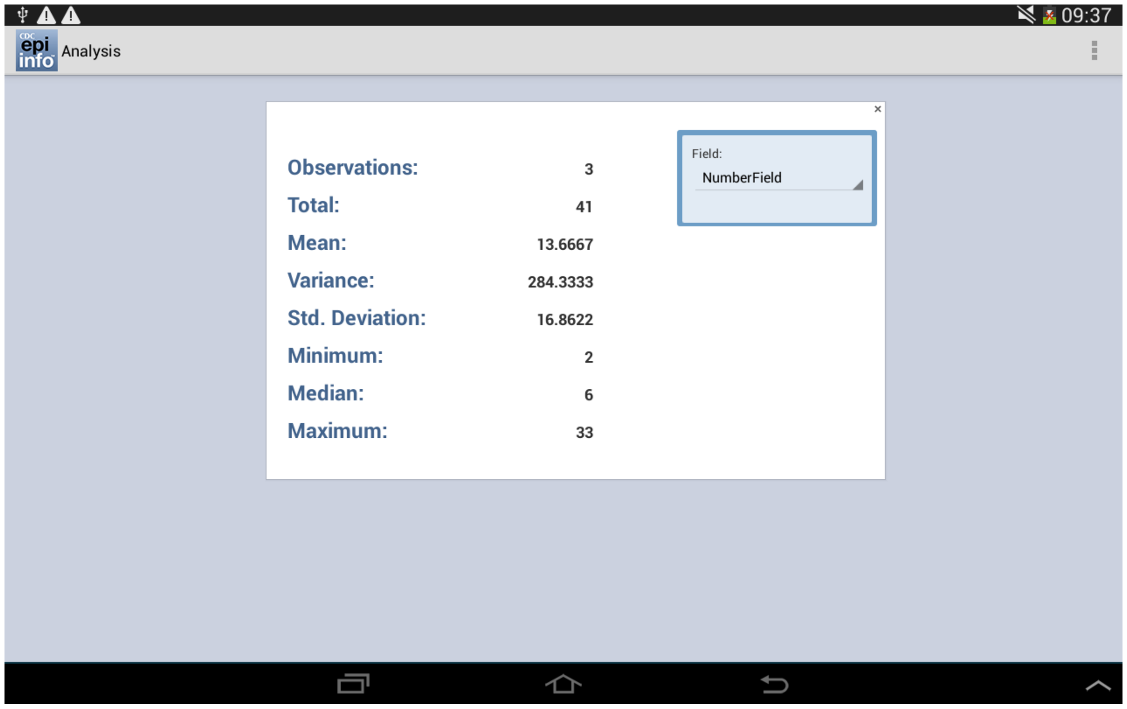
Phone

Figure 7.295: Analyze Data main menu

Note: The location of the options menu varies depending on the device.

Means Gadget

The Means gadget calculates the average for a continuous numeric variable. Mean, Variance, Standard deviation, Median, Minimum and Maximum values are calculated.


Analysis

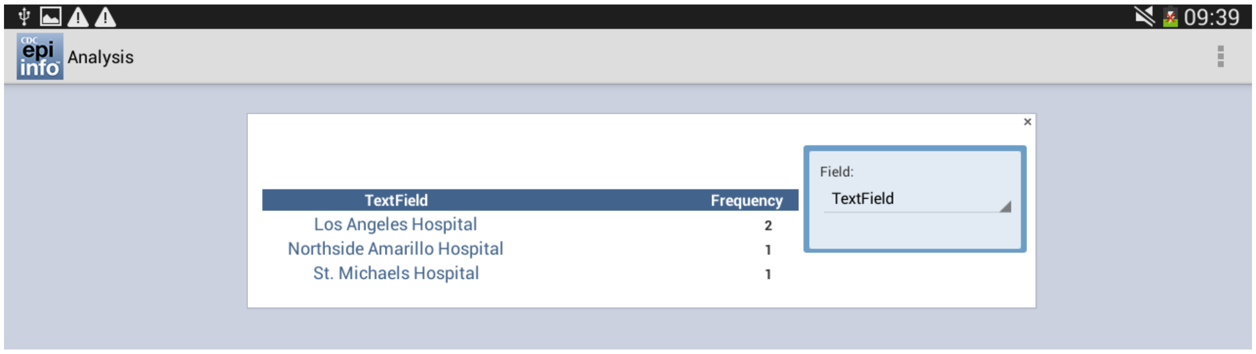
Observations:	3
Total:	41
Mean:	13.6667
Variance:	284.3333
Std. Deviation:	16.8622
Minimum:	2
Median:	6
Maximum:	33

Field: NumberField

Figure 7.296: Means Gadget

Frequency Gadget

The Frequency gadget counts each occurrence in a category for a specified variable and gives the absolute and relative frequencies for each category. This option then produces a frequency table that shows how many records have a value for each variable, the percentage of the total, a cumulative percentage and upper and lower confidence intervals.


Analysis

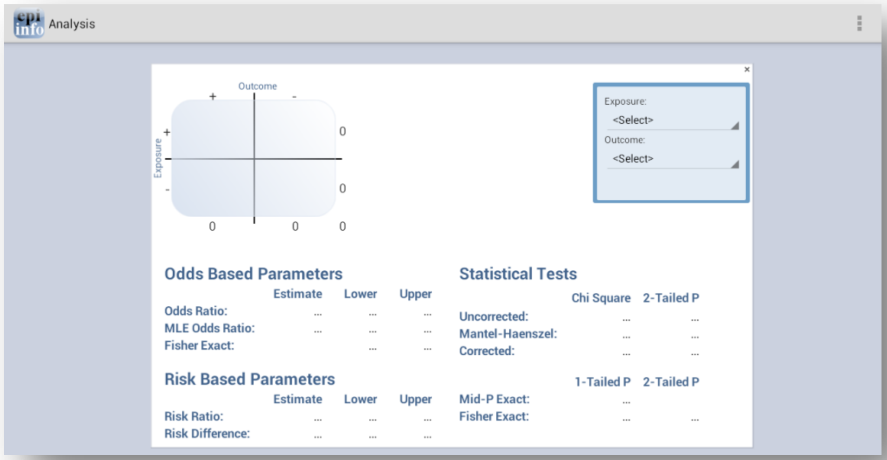
TextField	Frequency
Los Angeles Hospital	2
Northside Amarillo Hospital	1
St. Michaels Hospital	1

Field: TextField

Figure 7.297: Frequency Gadget

2x2 Gadget

In epidemiology, 2 x 2 tables are frequently used to examine the relationship between two or more categorical values. In these tables, usually an “exposure” variable is considered the risk factor. The outcome variable is considered the disease of consequence. (e.g. the person had the disease or outcome of interest or they did not). Values of the first variable will appear on the left margin of the table, and those of the second will be across the top of the table. Normally, cells contain counts of records matching the values in corresponding marginal labels.



Odds Based Parameters

	Estimate	Lower	Upper
Odds Ratio:	...	...	...
MLE Odds Ratio:	...	...	...
Fisher Exact:		...	...

Risk Based Parameters

	Estimate	Lower	Upper
Risk Ratio:	...	...	...
Risk Difference:	...	...	...

Statistical Tests

	Chi Square	2-Tailed P
Uncorrected:	...	...
Mantel-Haenszel:	...	...
Corrected:	...	...

	1-Tailed P	2-Tailed P
Mid-P Exact:	...	
Fisher Exact:	...	...

Figure 7.298: 2x2 Gadget

THIS PAGE IS

INTENTIONALLY BLANK.

7. Data Packager: Sharing and Merging Data

Introduction

The Epi Info™ Data Packager tool provides an easy way to share data with other users or to merge data collected by multiple users into a single database for analyses. The Data Packager does this by offering the option to package and export, as well as import data from Epi Info™ projects.

Data Packager also has several security features that allow data to be shared more safely and securely. One security feature is the capability to encrypt data packages with a password. This prevents anyone without the correct password from accessing the data. Another security feature is the capability to de-identify data by omitting specific fields from your database. Users can strip sensitive fields such as name, address, or ID number from the dataset. Users can also subset the data by specifying conditions for record selection. Data Packager also offers the option to save package settings for future use. Once packaged, the Import Data option easily imports the data.

To access Data Packager, select Enter Data from the Epi Info™ 7 main menu options and open a project and form. The Package for Transport option is located in the toolbar under **File > Package for Transport** and the Import Data from Data Package option is located under **File > Import data > From data package**.

Package Data for Transport

An Epi Info™ project must be opened with the Enter tool before the data in that project can be packaged for transport.

1. From the Enter screen, open the project and form you wish to transport. If you have related forms in your project, they will not appear in the Project Explorer window of the Enter tool. However, all the data in child forms will be packaged for transport.
2. Select **File > Package for Transport**. The Package data for Transport dialog box appears.

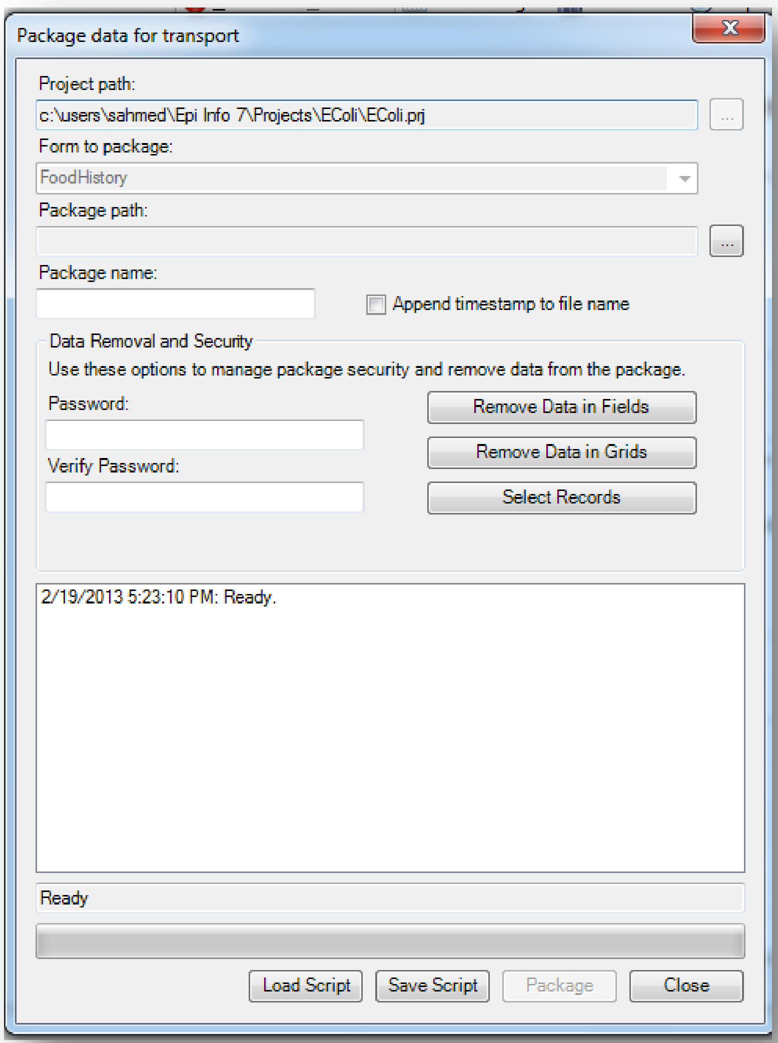
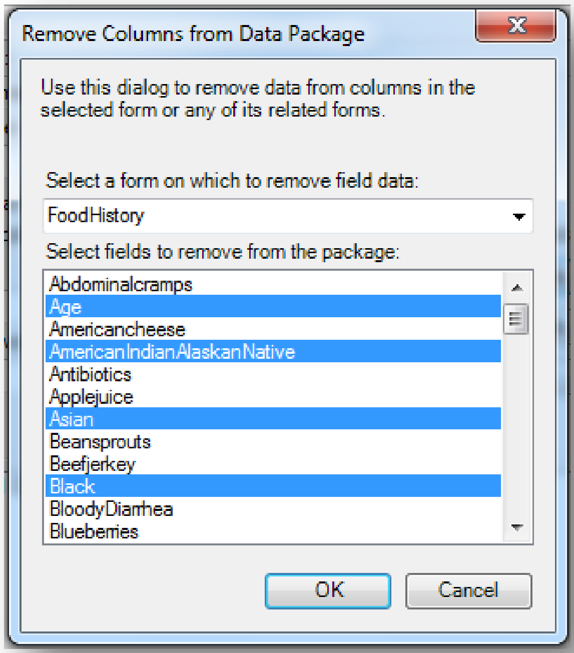

Package data for transport
Project path:
c:\users\sahmed\Epi Info 7\Projects\EColi\EColi.prj
Form to package:
FoodHistory
Package path:
Package name:
Append timestamp to file name
Data Removal and Security
Use these options to manage package security and remove data from the package.
Password:
Verify Password:
Remove Data in Fields
Remove Data in Grids
Select Records
2/19/2013 5:23:10 PM: Ready.
Ready
Load Script
Save Script
Package
Close

Figure 7.1: Package Data for Transport dialog box

3. The **Project path** and **Form to package** textboxes automatically populate with the project and form information. Click the **ellipses** to browse and select a **Package Path**. The package path designates where the data package will be saved. Click **OK**.
4. Enter a name for the data package in the **Package Name** field.
5. Check the **Append Timestamp to File Name** checkbox if you wish to add a timestamp to the file name. The default setting is unselected.
6. **Enter Password** if you wish to encrypt the data package. This increases the security of your data transmission. Save your password in a safe location, as this password will be required to access the data package. If you do not wish to encrypt your data, leave the Enter Password field blank and proceed to step eight.
7. To **Verify Password**, re-enter the password from the step above.

There are several options to remove data prior to packaging: Remove Data in Fields, Remove Data in Grids, and Select Records.

8. Click the **Remove Data in Fields** button to omit specific fields from the data package.
   1) **Select a form on which to remove field data** from the drop-down list.
   2) **Select fields to remove from the package** by clicking on the field names. To select more than one field, hold down the CTRL key.
   3) Click **OK**. The selected fields will not be included in the packaged dataset.


Remove Columns from Data Package

Use this dialog to remove data from columns in the selected form or any of its related forms.

Select a form on which to remove field data: FoodHistory

Select fields to remove from the package:

Abdominalcramps
Age
Americancheese
AmericanIndianAlaskanNative
Antibiotics
Applejuice
Asian
Beansprouts
Beefjerkey
Black
BloodyDiarrhea
Blueberries

Figure 7.2: Remove Fields from Data Package dialog box

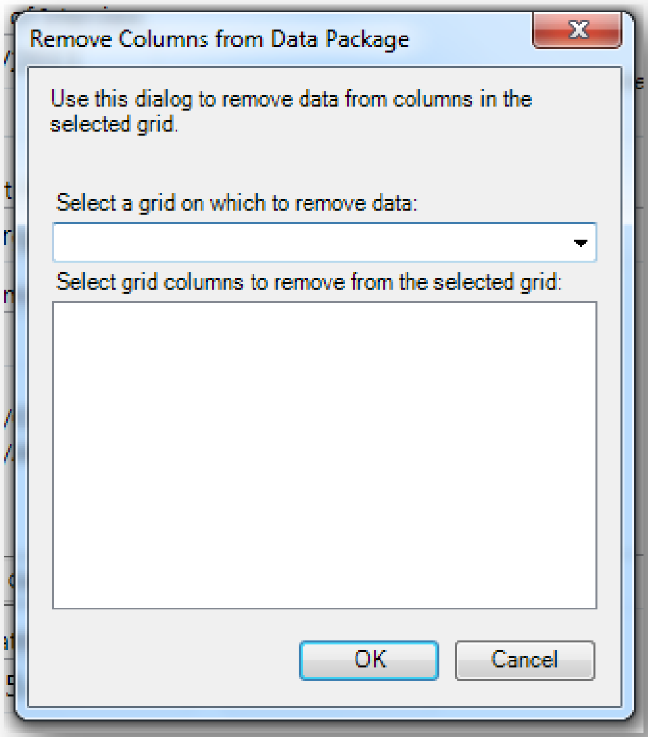
9. Click the **Remove Data in Grids** button to remove data from columns in the selected grid.

    1) **Select a grid on which to remove data** from the drop-down list.

You will only see options in the drop-down list if the form you are packaging contains grid fields. The items in the drop-down list will represent all grids that exist on the form and any descendant forms, and the items in the list box will represent the grid columns in the chosen grid field.

    2) **Select grid columns to remove from the selected grid** from the drop-down list.

    3) Click **OK.** The selected fields will not be included in the packaged dataset.


Remove Columns from Data Package

Use this dialog to remove data from columns in the selected grid.

Select a grid on which to remove data:

Select grid columns to remove from the selected grid:

Figure 7.3: Remove Grids from Data Package dialog box

10. Click the **Select Records** button to select specific records to include in the data package.

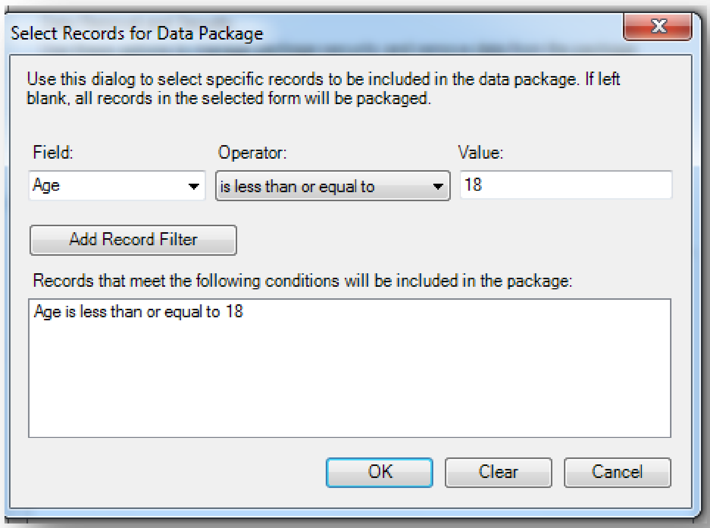
    1) Select a **Field** from the drop-down list.

    2) Select an **Operator** from the drop-down list.

    3) Enter a value in the **Value** textbox.

4) Click the **Add Record Filter** button. The specified conditions appear in the textbox. All records that meet this condition will be included in the data package. All other records will be omitted.

5) Click **OK.**


Select Records for Data Package

Use this dialog to select specific records to be included in the data package. If left blank, all records in the selected form will be packaged.

Field: Age
Operator: is less than or equal to
Value: 18

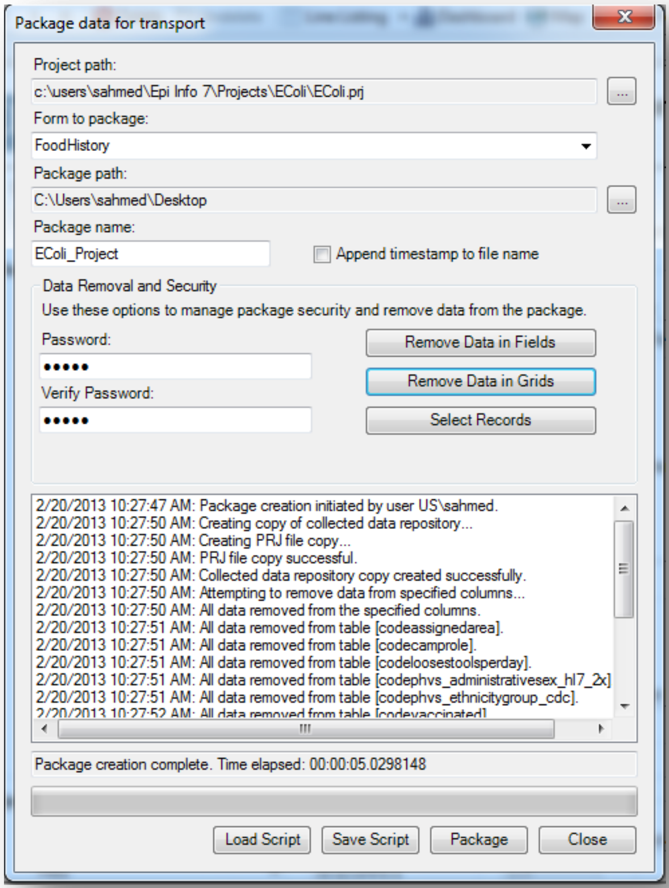
Records that meet the following conditions will be included in the package:

Age is less than or equal to 18

Figure 7.4: Select Records for Data Package dialog box

11. The **Package data for Transport** dialog box appears.

12. Click the **Package** button. When complete, the *Package Creation Complete* verbiage will appear at the bottom of the dialog box.



Project path:
c:\users\sahmed\Epi Info 7\Projects\EColi\EColi.prj
Form to package:
FoodHistory
Package path:
C:\Users\sahmed\Desktop
Package name:
EColi_Project
Append timestamp to file name

Data Removal and Security
Use these options to manage package security and remove data from the package.
Password:
•••••
Verify Password:
•••••

```
2/20/2013 10:27:47 AM: Package creation initiated by user US\sahmed.
2/20/2013 10:27:50 AM: Creating copy of collected data repository...
2/20/2013 10:27:50 AM: Creating PRJ file copy...
2/20/2013 10:27:50 AM: PRJ file copy successful.
2/20/2013 10:27:50 AM: Collected data repository copy created successfully.
2/20/2013 10:27:50 AM: Attempting to remove data from specified columns...
2/20/2013 10:27:50 AM: All data removed from the specified columns.
2/20/2013 10:27:51 AM: All data removed from table [codeassignedarea].
2/20/2013 10:27:51 AM: All data removed from table [codecamprole].
2/20/2013 10:27:51 AM: All data removed from table [codeloosestoolsperday].
2/20/2013 10:27:51 AM: All data removed from table [codephvs_administrativesex_hl7_2x]
2/20/2013 10:27:51 AM: All data removed from table [codephvs_ethnicitygroup_cdc].
2/20/2013 10:27:52 AM: All data removed from table [codevaccinated]
```

Package creation complete. Time elapsed: 00:00:05.0298148

Figure 7.5: Package Creation complete

Additional Options:

Note that the textbox on the bottom half of the screen is populated with your Data Removal and Security selections. You may save this script to be used at a future time. This will prevent you from having to make each selection manually the next time you wish to package data.

a. The **Save Script** option will save your settings for future use. Note: The password is not saved in this script.

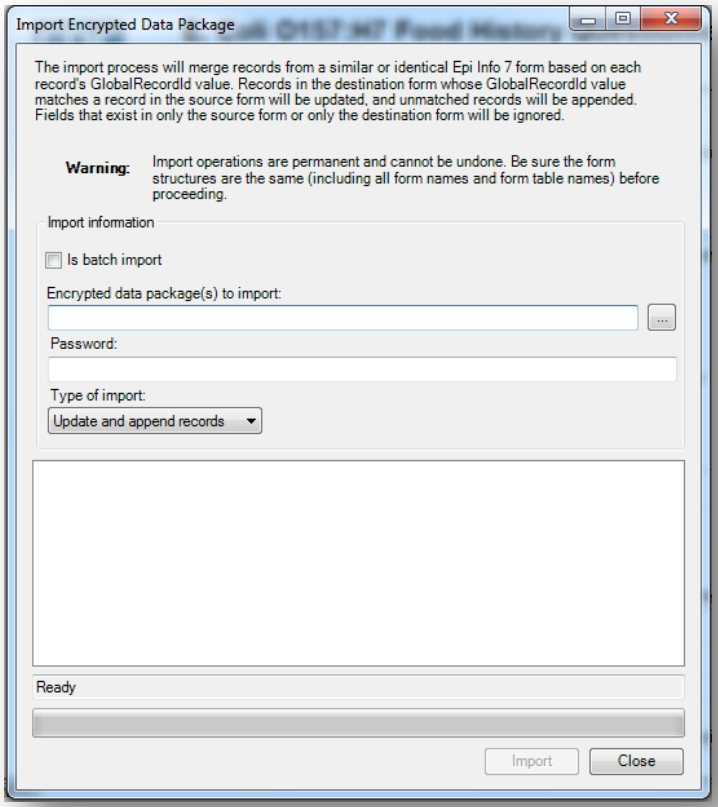
b. The **Load Script** option imports and replicates the saved script containing your package settings.

13. Click **Close** to exit.

14. The data package appears at the location (Package Path) specified in step three.

Import Data from Data Package

7. From the Enter screen, select **File > Import Data > From Data Package.**


Import Encrypted Data Package
Import information
Is batch import
Encrypted data package(s) to import:
...
Password:
Type of import:
Update and append records
Ready
Import
Close

The import process will merge records from a similar or identical Epi Info 7 form based on each record's GlobalRecordId value. Records in the destination form whose GlobalRecordId value matches a record in the source form will be updated, and unmatched records will be appended. Fields that exist in only the source form or only the destination form will be ignored.

Warning: Import operations are permanent and cannot be undone. Be sure the form structures are the same (including all form names and form table names) before proceeding.

Figure 7.6: Import Encrypted Data Package dialog box

8. Check the **Is Batch Import** checkbox if attempting to import more than one *.edp7 file. When selected, the batch import will only prompt for the folder location containing the *.edp7 files. All files in the selected folder will be imported at once. This option is unchecked by default.

9. Click the **ellipses** to browse the location for the **Encrypted data package(s) to import**.

10. Select the location of your *.edp7 file. Click **Open**.

11. If the data package is encrypted, enter the password in the **Password** textbox. This password was created when the data was initially packaged. If the data is not encrypted with a password, leave the password textbox blank and continue to the next step.

Note: When accessing an encrypted data package, if the password is incorrect or left blank, the message below will appear at the bottom of the dialog box. In addition, when using batch import, all files in a directory must contain the same password or you will get the error message below:

Import FAILED. The package was unable to be decrypted. Check to make sure the password is correct and try again.

12. Select the **Type of Import** from the drop-down list.

    - **Update and append records** – Records in the destination form containing the same GlobalRecordId values as the source form will be updated (overwritten) if there is new information. However, values in the destination table will never be overwritten with a null or missing value. All other records will be appended (added) to the end of the table.
    - **Update records only** – Records in the destination form containing the same GlobalRecordId values as the source form will be updated (overwritten) if there is new information. However, values in the destination table will never be overwritten with a null or missing value. All other records will be ignored.
    - **Append records only** – All records in the source form will be appended (added) to the end of the table. No records will be updated (overwritten).

Note: Import operations are permanent and irreversible. Be sure that the form structures are the same (including all form names and form table names) before proceeding.

13. Click **Import**. *Import Complete* will appear at the bottom of the dialog box upon successful completion of the import.

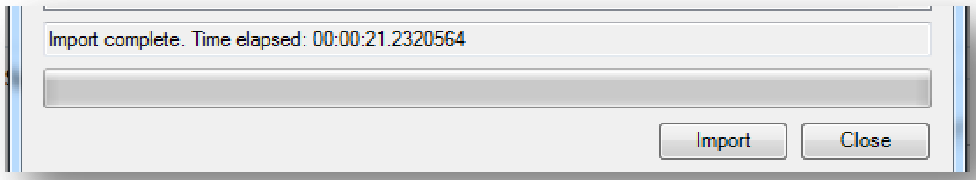

Import complete. Time elapsed: 00:00:21.2320564

Figure 7.7: Import Complete

14. Click **Close** to exit the dialog box. The imported data appears in the form on the Enter screen.

8. Visual Dashboard: Performing Statistical Analyses with Visual Tools

Introduction

Visual Dashboard is one of the Epi Info™ 7 analysis modules. The Visual Dashboard is designed to be intuitive and simple to use. With the use of gadgets, the need for programming code is minimized. Data can be selected, sorted, listed or manipulated with the various gadgets in Visual Dashboard. The statistical analyses tools available in Visual Dashboard include frequencies, means, and more advanced statistical calculations processes (e.g., linear regression and logistic regression). Visual Dashboard has graphing functionality to display data as an Epi Curve, Pareto Chart, and several other bar and column charts.

Visual Dashboard can be accessed from the Epi Info™ 7 main window by clicking the Visual Dashboard button or by selecting **Tools > Analyze Data>Visual Dashboard** from the main page navigation menu. It can also be accessed from the toolbar menu of the Enter tool once data has been loaded into Epi Info™ 7.

Navigate the Visual Dashboard Workspace

The Visual Dashboard workspace contains four main areas: the canvas, Record Count/Data Source toolbar, Data Recoding and Formatting gadget (sometimes referred to as the Defined Variables gadget) and the Data Filters gadget.

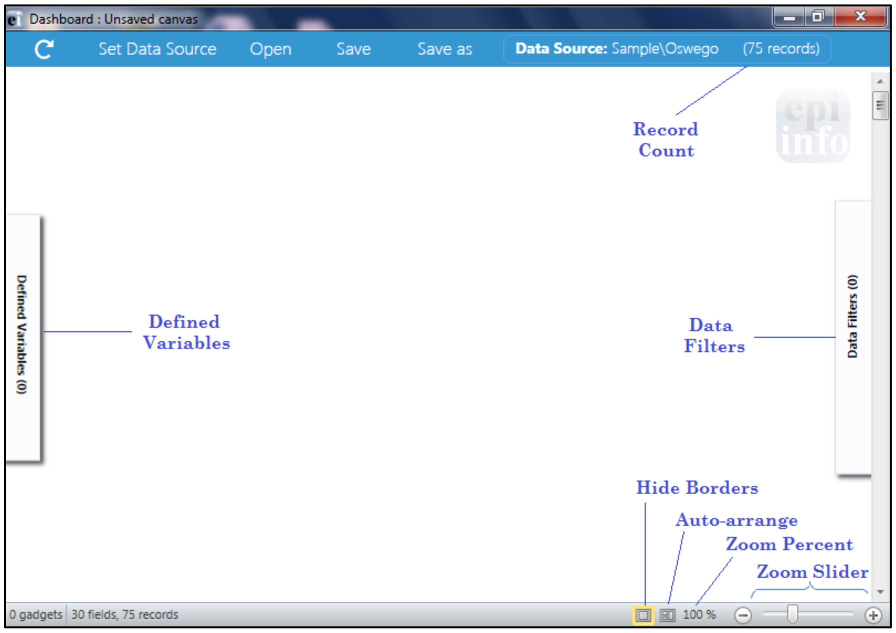


Figure 7.8: Visual Dashboard canvas

The **Visual Dashboard canvas** displays output generated by the multiple gadgets available in the tool. At the bottom right side of the Dashboard, the Hide Borders button turns off the external borders for each gadget giving the canvas a more report-like appearance. The Auto-arrange button positions the gadgets on the canvas to maximize the display area. The Zoom Percent and Zoom Slider allow you to zoom in and out of the canvas. The gadgets that get loaded automatically after a dataset has been uploaded are:

1. The **Record Count/Data Source**, located in the toolbar at the top of the screen, displays the source and record count of the data being analyzed. The number of fields (columns) in your dataset will be displayed on the bottom left of the status strip.
2. The **Data Recoding and Formatting Gadget (Defined Variables**), allows for the creation, editing, and deletion of variables created during the analysis session. This gadget is located on the left-hand side of the screen.
3. The **Data Filtering Gadget (Data Filters)**, which is a canvas wide filter, is on the right-hand side of the screen and used to select a subset of data for analysis.

Gadget Filters and Settings

In addition to setting canvas wide filters, you can also set a gadget-specific filter using the widgets at the top of the gadget. The widgets and borders disappear when the "Hide Borders" button is pressed.

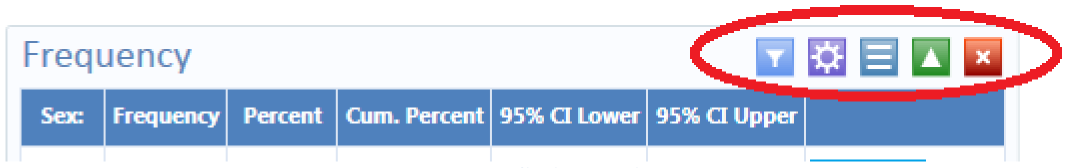


Figure 7.9: Gadget widgets

1. The funnel icon on the gadget's header panel works the same way as the canvas-wide filters (Defined Variables gadget and Data Filters gadget), except that: (1) Advanced mode is not available, and (2) the filtering applies only to that gadget. An important point to note is that gadget filters are combined with canvas filters. E.g. if the canvas filter is 'Age > 20' and the gadget filter is 'ILL = Yes', then the filter applied to the gadget is 'Age >20 AND ILL = Yes'. Some gadgets do not have filtering capabilities, such as the advanced statistics.
2. The purple button collapses and expands the gadget's properties.
3. The middle button (with the three lines) on the gadget header panel sets a gadget's title and description. The title and description are displayed when the dashboard is exported in HTML format.
4. The green up/down arrow on the gadget's header panel collapses and expands the gadget's output, which is useful when you want to minimize the amount of gadgets that are displayed on the canvas.
5. You may exit the gadget by clicking on the red "X" at the top right of the header panel.

Canvas Properties

Canvas properties, which can be accessed by right clicking on the canvas and selecting **Canvas Properties** from the menu options, are used to:

- Change the data source properties by selecting a new project and table
- Change HTML output properties and create a custom title, summary, and conclusion for your Visual Dashboard output
- View miscellaneous information about your data such as number of rows and columns, and cached information

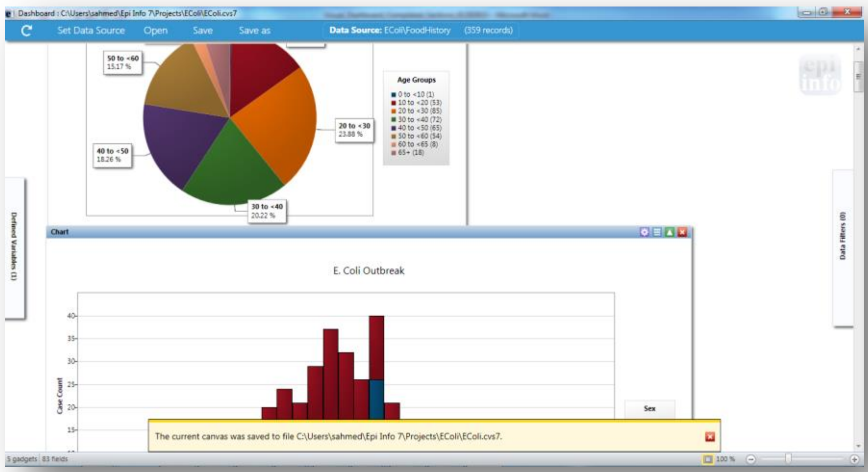
Note: Changing the data source is not advisable except for advanced users.

Save and Open a Canvas

The current canvas including data source, gadgets, filters, and user-defined variables may be saved and reused later. Saved canvases may be opened later with the Visual Dashboard tool. This will allow you to reconnect to the data source without having to manually apply the filters again.

To save a canvas, right click on the canvas and select **Save Canvas**.

- If the canvas was previously saved and named, this will overwrite the existing file.
- A canvas being saved for the first time will follow the process outlined in the 'Save As' option. Epi Info™ 7 requires a "save to" location and a file name.



The current canvas was saved to file C:\Users\sahmed\Epi Info 7\Projects\EColi\EColi.cvs7.

Figure 7.10: Saving Canvas

The yellow message at the bottom of the screen confirms the canvas was saved successfully.

Save Canvas As

The **Save Canvas As** option allows the canvas to be saved for the first time or saved to a new location.

1. Right click on the canvas, select **Save Canvas As**. The **Save As** dialog box appears.
2. Select a location to save the canvas to and enter a File Name.
3. Click **Save**.

Note: If a file with the same name already exists, an error message will appear asking if you would like to replace existing file.

Open Canvas

Open an existing Visual Dashboard canvas by performing the following steps:

1. Right click on the canvas and select **Open Canvas**. The **Open Canvas** dialog box appears.

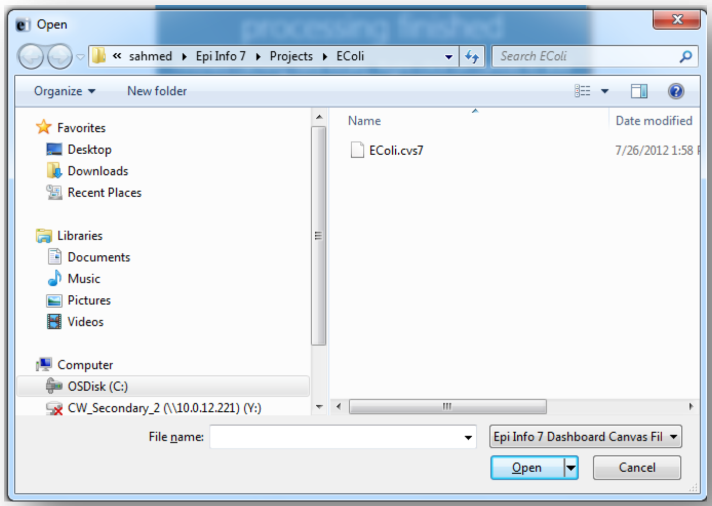
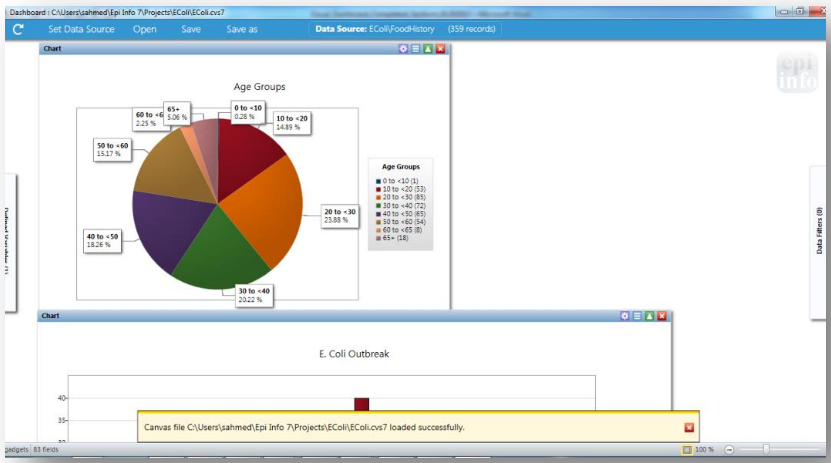


Figure 7.11: Open a Canvas

2. Select the location of the canvas file. The file can be identified by the file name extension (**.cvs7**).
3. Click **Open.** The canvas file appears in Visual Dashboard.



Canvas file C:\Users\sahmed\Epi Info 7\Projects\EColi\EColi.cvs7 loaded successfully.

Figure 7.12: Opened Canvas

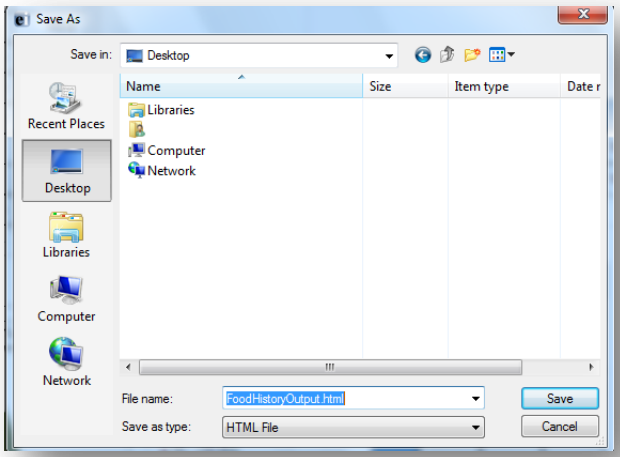
The yellow message at the bottom of the screen confirms the canvas loaded successfully.

Save Visual Dashboard HTML

Output from the gadgets may be saved as an HTML document. To save the current output:

1. Right click on the canvas, select **Save Output as HTML**. The **Save As** dialog box appears.

2. Select a location to save the canvas and enter a File Name. Select **HTML File** from the **Save as Type** drop-down list.



File name: FoodHistoryOutput.html
Save as type: HTML File

Figure 7.13: Save Output as HTML

3. Click the **Save** button. The output automatically opens in the web browser.

Note: HTML documents can only be opened in a web browser and cannot be used to re-create the Visual Dashboard canvas.

Send Visual Dashboard Output To

Outputs from Visual Dashboard may be viewed in either Microsoft Word or Microsoft Excel with the **Send Output To** option. You must have Microsoft Word or Microsoft Excel installed to use either option.

Microsoft Excel

1. Right click on the canvas, select **Send Output To > Microsoft Excel**.
2. The **Visual Dashboard** output opens in **Microsoft Excel**.

Note: Charts are not included with the output when sent to Excel.

Microsoft Word

1. Right click on the canvas, select **Send Output To > Microsoft Word**.
2. The **Visual Dashboard** output opens in **Microsoft Word** as an .html formatted web document. If saved in this format, charts are saved separately to a sub-folder. To share this .html file, remember to share the subfolder as well. Alternatively, change the format to a Microsoft Word .docx format so the charts are included with the file as embedded images.

Managing Data

Selecting a Data Source

Before beginning analysis in Visual Dashboard, select a data source. Data from the following sources can be analyzed in Visual Dashboard: Epi Info™ 7 project files, ASCII Text files, Microsoft Excel, Microsoft Access, Microsoft SQL Server, MySQL, and PostgreSQL. Follow one of the first three steps below to select a data source:

1. Click on the **Set Data Source** option from the bar at the top of the screen.

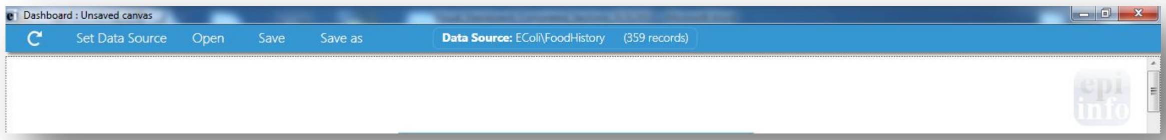
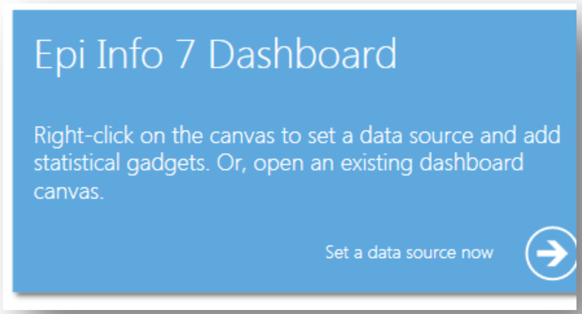

Dashboard : Unsaved canvas
Set Data Source
Open
Save
Save as
Data Source: ECoil\FoodHistory (359 records)

Figure 7.14: Visual Dashboard toolbar

2. Click the arrow in the blue box next to **Set a Data Source Now**.



Epi Info 7 Dashboard

Right-click on the canvas to set a data source and add statistical gadgets. Or, open an existing dashboard canvas.

Figure 7.15: Set Data Source option

3. Right-click on the canvas and select **Set Data Source** from the menu options.
4. After selecting the **Set Data Source** option, the **Select Data Source** dialog box appears. Select a **Database Type**, **Data Source**, and form to open data in Visual Dashboard.

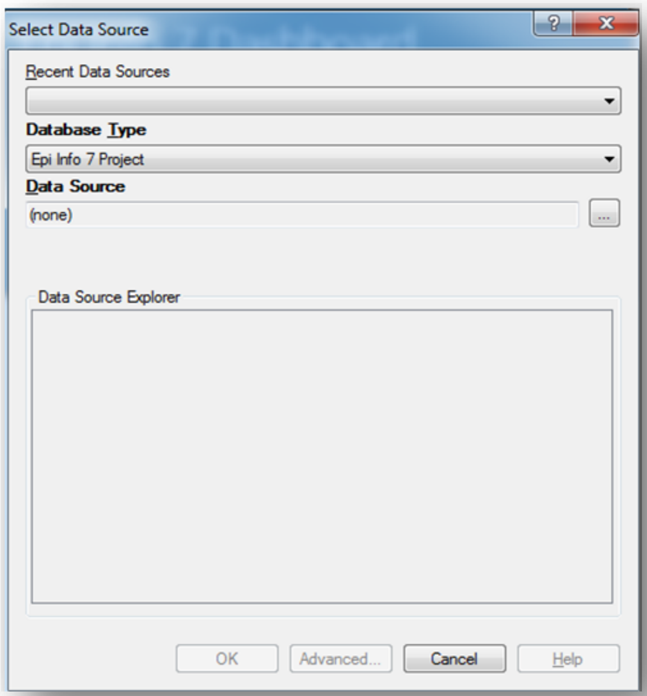


Figure 7.16: Select Data Source dialog box

- The **Recent Data Sources** field provides a list of recently accessed databases in Classic Analysis or Visual Dashboard. By selecting one of the data sources available on the list, you do not need to provide Database Type or Data Source information. If you are reading an Epi Info™ 7 project file from the Data Source section, all available forms will appear in the Data Source Explorer box.
- The **Database Type** field indicates the database file to be loaded (.PRJ, .MDB, .XLS). Specify the data format for the data being accessed. Unless otherwise specified, output files are stored in this destination folder as well.
- The **Data Source** field indicates the file location/path. If you use an SQL Server database, the Data Source field will require server and database names.
- The **Data Source Explore**r field allows you to select a form in your project. This reads the data table associated with your form.
- The **Advanced** mode allows the use of SQL queries to select the data source. Only SELECT statements are allowed.

Read an Epi Info™7 Project File

1. Click on the **Set a Data Source now** arrow. The **Select Data Source** dialog box appears.
2. From the **Database Type** drop-down list, select **Epi Info 7 Project.**

3. In the **Data Source** field, click on the **ellipsis** button and the **Epi Info 7 Project** folder appears.
4. Browse for the project file that you would like to import into Visual Dashboard.
5. Click **Open.**
6. The **Data Source Explorer** box populates with a list of all the forms that are associated with the project. Select a form.
7. Click **OK**. The record count and data source appear in the tool bar at the top of the Visual Dashboard canvas.

Read a Microsoft Excel File

1. Click on the **Set a Data Source now** arrow. The **Select Data Source** dialog box appears.
2. From the **Database type** drop-down list, select either **MS Excel 97-2003 Workbook** or **MS Excel 2007 Workbook.**
3. In the **Data Source** field, click on the **ellipsis** button and the **Open Existing File** dialog box appears.
4. Enter the filename and path to open your workbook or click on the **ellipsis** button to browse for the **MS Excel workbook** (.xls or .xlsx).
5. Click **Open.** The "First row contains Field Names" checkbox is selected by default. If this option is unchecked, data will be imported without column headings.
6. Click **OK**. Select a **worksheet** from the list provided in the Data Source Explorer.
7. Click **OK**. The record count and data source appear in the menu bar at the top of the Visual Dashboard canvas.

Read a Microsoft Access File

1. Click on the **Set a Data Source now** arrow. The **Select Data Source** dialog box appears.
2. From the **Database Type** drop-down list, select either **MS Access 2002-2003 (.mdb)** or **MS Access 2007(.accdb).**
3. In the **Data Source** field, click on the **ellipsis** button and the **Open Microsoft Access File** dialog box appears.
4. Enter the filename and path to open your file or click on the **ellipsis** button to browse for the **MS Access file (.mdb or .accdb)**. If the file is password protected, enter the password.
5. Click **OK.**
6. Select a **data table** from the list provided in the **Data Source Explorer**.
7. Click **OK**. The record count and data source appear in the menu bar at the top of the Visual Dashboard canvas.

Read an ASCII Text File

1. Click on the **Set a Data Source now** arrow. The **Select Data Source** dialog box appears.
2. From the **Database Type** drop-down list, select **Flat ASCII File**.
3. In the **Data Source** field, click the **ellipsis** button to browse.
4. Click the **ellipsis** button in the **Flat File** location dialog box to browse the directory of the ASCII file
5. Click **OK**. Note that the First row contains header information box is checked by default. If the first row of the file does not contain header information, deselect the checkbox. If this option is unchecked, data will be imported without column headings.
6. Select the **file name** from the list provided in the **Data Source Explorer**.
7. Click **OK.** The record count and data source appear in the menu bar at the top of the Visual Dashboard canvas.

Read a SQL Server Database

Note: You must have Microsoft SQL Server installed on your computer or you must have access to a server over a network to perform this function.

1. Click on the **Set a Data Source now** arrow. The **Select Data Source** dialog box appears.
2. From the **Database Type** drop-down list, select **Microsoft SQL Server Database**.
3. In the **Data Source** field, click the **ellipsis** button to browse.
4. The **Connect to SQL Server** database dialog box opens. Specify the **Server name** and **Database name** to connect to a SQL Server database.
5. The Use Windows Authentication radio button is selected by default. You can use **SQL Server Authentication** by checking the corresponding radio button and then entering Login and Password credentials.
6. Click **OK**.
7. From the list provided in the **Data Source Explorer**, select a **data table**.
8. Click **OK.** The record count and data source appear in the menu bar at the top of the Visual Dashboard canvas.

Read a MySQL Server Database

Note: You must have MySQL Server installed on your computer or you must have access to a server over a network to perform this function.

1. Click on the Set Data Source arrow. The Select Data Source dialog box appears.
2. From the **Database Type** drop-down list, select **MySQL Database**.

3. In the **Data Source** field, click the **ellipsis** button to browse SQL databases.
4. The **Connect to MySQL** database dialog box appears. Enter the **Server name**, **Database name**, **Username** and **Password** to connect to the MySQL database. You can select **<Browse for more>** from the **Server name** drop-down list to select a MySQL server instance in your network.
5. Click **OK**.
6. From the list provided in the Data Source Explorer, select a **data table**.
7. Click **OK.** The record count and data source appear in the menu bar at the top of the Visual Dashboard canvas.

Add a Related Data Source

Analysis in Visual Dashboard may be performed on data from multiple data sources. The **Add a Related Data Source** option links one or more tables using a common identifier to find matching records. The form/table to be linked requires a key field that relates records in the two forms/tables together. The keys in the main and related tables or forms do not require the same name. If the table was created in Form Designer and the data entry was completed using Enter, Visual Dashboard can establish a relationship by using the Global Record ID and Foreign Key variables created by Epi Info™ 7. The example below demonstrates how to add a related data source.

Epi Info™ 7 contains a related database in the Sample.PRJ file with two forms: Surveillance and RHepatitis. The variable GLOBAL RECORD ID is the internal Epi Info™ 7 identification key located in more than one form in the project.

1. Select the **Sample.PRJ** Data Source. Open **Surveillance**.
2. Right click on the canvas and select **Add related data source**. The **Select Data Source** dialog box opens.
3. Select **RHepatitis** from the list of forms in the Data Source Explorer. The **Child Key Field** drop-down list populates.
4. From the **Parent Key Field** drop-down list, select **GLOBALRECORDID**.
5. From the **Child Key Field** drop-down list, select **FKEY**.
6. Click **OK**. The related form information appears on the canvas. Data from the two tables can now be used for analysis.

Note: You can only perform a one-to-one analysis with a related table in Visual Dashboard. To perform a one-to-many analysis, please see the Classic Analysis section of this user guide.

Managing Variables

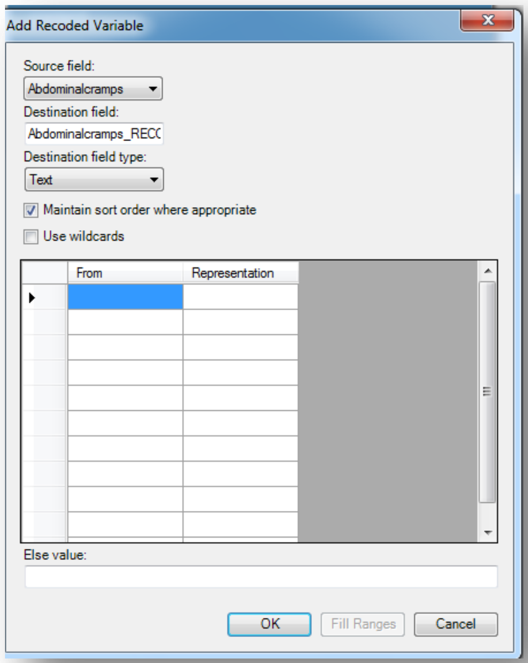
The Data Recoding and Formatting Gadget in Visual Dashboard allow you to create, edit, or delete variables with no coding. To access the Data Recoding and Formatting gadget, move your mouse over Defined Variables on the left-hand side of the screen.

Creating a New Variable

Project data are not always in the proper format needed for analysis. As an example, a patient's age may be entered in a questionnaire but the ages may need to be in five-year increments for proper analysis. The data may contain county codes, but descriptive county names may be more useful for analysis. Creating a new variable is one of the functions of the Data Recoding and Formatting gadget. By creating a new variable, the ages of patients can be recoded to five-year increments, or county names may be assigned to the county codes making the data useful for analysis. A new variable may be created to change the format of existing data with recoded value, simple assignment, formatted value, assigned expression, or by grouping existing variables.

Recoded Value

The New Variable with Recoded Value option assigns a new value to existing data and places the results in defined variables. The **Add Recoded Variable** dialog box is pictured below.


Add Recoded Variable

Source field: Abdominalcramps

Destination field: Abdominalcramps_RECC

Destination field type: Text

[✓] Maintain sort order where appropriate

[ ] Use wildcards

From	Representation

Else value:

Figure 7.17: Add Recoded Variable Options

- The **Source field** drop-down list represents the column in the current data source that will have its values recoded.
- The **Destination field** drop-down list represents a new column that will be temporarily created in the current data source that will store the recoded values.
- The **Destination field type** represents the data type of the new column that will be created to store the recoded values. The available options are text, numeric, or Yes/No.
- The **Maintain Sort Order Where Appropriate** is checked by default. Usually output in Visual Dashboard is sorted alphabetically or sequentially. Leave this box checked if you would like the output to remain sorted exactly in the sequence that the data was entered.
- The **Use Wildcards** option can be used for text searches in the FROM column below. This option is unchecked by default. The asterisk (*) is the wildcard character.
- The **From** column identifies the bottom of a range of values or a single value. To recode a single value instead of a range, use this From column only and ignore the To column. When used to recode a range of values, the recode range includes the value in the From column and extends up to, but not including the value in the To column. To recode all unspecified values, leave both From and To columns blank.
- **To** value (if any) identifies the upper limit of a range of values to be recoded. The recode range does not include the value in the To column. This option only appears if a numeric variable is selected.
- The **Else Value** represents an optional value to assign to any records that do not match the recoding rules above. If left blank, unmatched records will be left null.
- **Representation Value** identifies the value to be assigned to the destination variable for the specified values of the source variable.
- **Fill Ranges** can be used to automatically populate range values on a numeric source field and is a very quick way to create numeric groupings such as age categories.
- **OK** accepts the current settings and data, and closes the form or window.
- **Cancel** exits the window without saving or executing a command.

The example below demonstrates how to add a recoded variable:

In the E. coli Food History table, there are over 60 different values for Age. When performing an analysis, it is difficult to compare all of the different age values. By grouping the age values in 10-year increments, analysis on this variable is simplified. To group the values in Age groupings or categories:

1. Select the **EColi.PRJ** Data Source.
2. From the Form section, click **FoodHistory**. Click **OK**.

3. On the left-hand side of the Visual Dashboard canvas, move the mouse cursor over the **Defined Variables** gadget. The gadget expands and becomes fully visible.
4. Click the **New Variable** button.
5. Select **With Recoded Value**. The **Add Recoded Variable** dialog box appears.

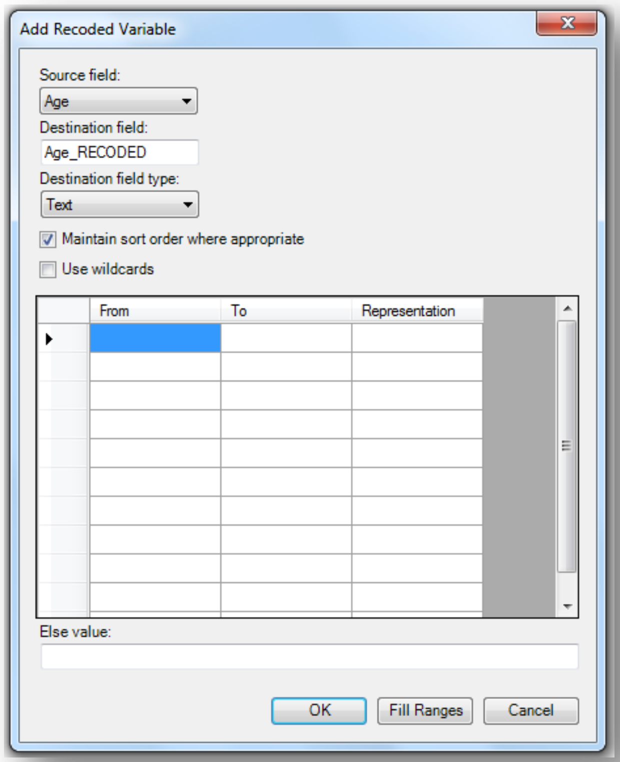
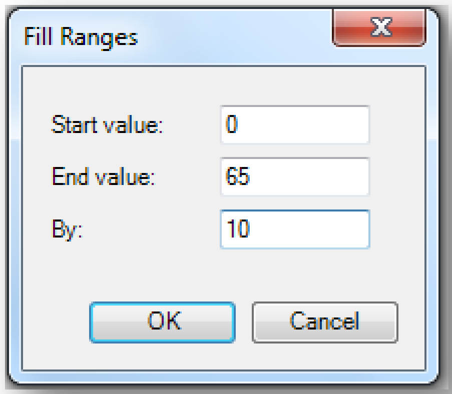


Figure 7.18: Add Recoded Variable dialog box 2

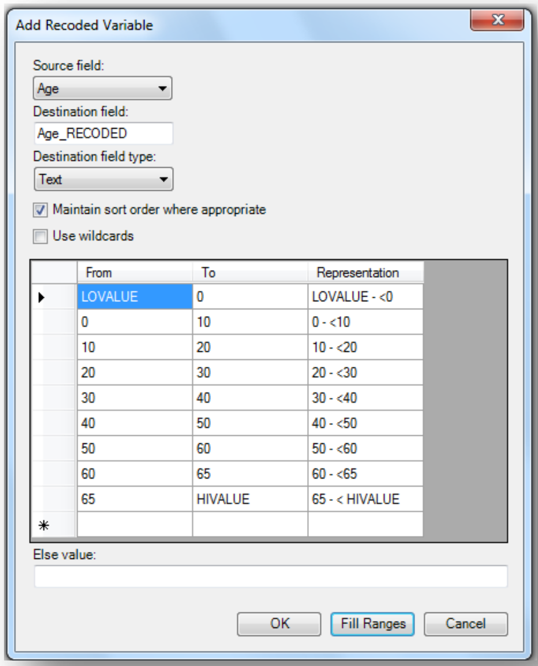
6. From the **Source field** drop-down list, select **Age**.
7. Click the **Fill Ranges** button. The **Fill Ranges** dialog box appears.
8. Enter **0** for the **Start value**, **65** for the **End value**, and **10** for **By**.



Start value: 0
End value: 65
By: 10

Figure 7.19: Add Recoded Variable/Fill Ranges

9. Click **OK**. The **Add Recoded Variable** dialog box appears with the **From**, **To**, and **Representation** columns populated.



Source field: Age
Destination field: Age_RECODED
Destination field type: Text
[✓] Maintain sort order where appropriate
[ ] Use wildcards

From	To	Representation
LOVALUE	0	LOVALUE - <0
0	10	0 - <10
10	20	10 - <20
20	30	20 - <30
30	40	30 - <40
40	50	40 - <50
50	60	50 - <60
60	65	60 - <65
65	HIVALUE	65 - < HIVALUE

Else value:

Figure 7.20: Add Recoded Variable Ranges Populated

10. Click **OK**. The variables are recoded and the conditions appear in the **Variable Definitions** textbox.

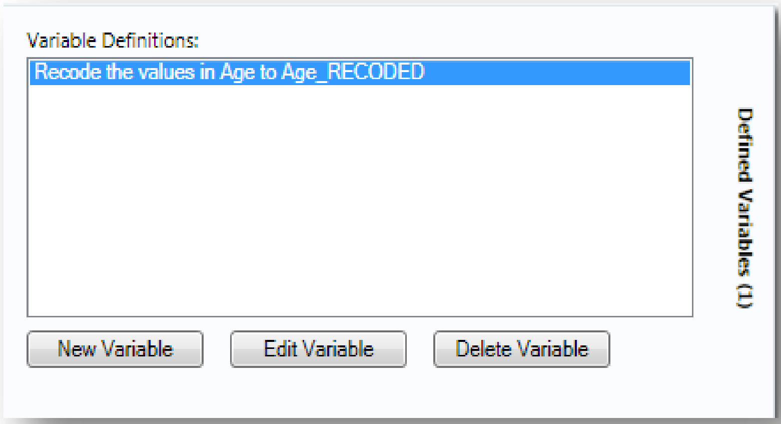
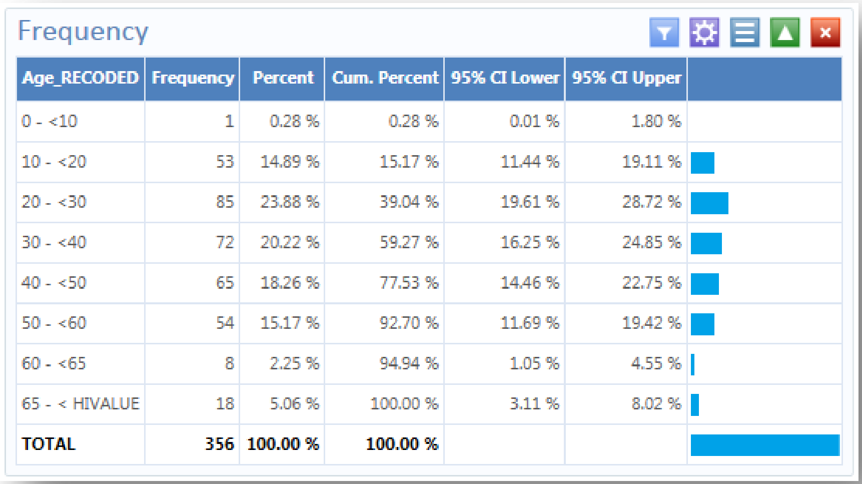


Figure 7.21: Variable Definitions dialog box

To see the frequency of Age_RECODED in line list view, right click on the canvas, and then select **Add Analysis Gadget > Frequency**. Select **Age_RECODED** from the **Frequency of** drop-down list and then click **Run.** For information regarding the Analysis Gadget, refer to the Displaying Statistics and Records section.


Frequency

Age_RECODED	Frequency	Percent	Cum. Percent	95% CI Lower	95% CI Upper
0 - <10	1	0.28 %	0.28 %	0.01 %	1.80 %
10 - <20	53	14.89 %	15.17 %	11.44 %	19.11 %
20 - <30	85	23.88 %	39.04 %	19.61 %	28.72 %
30 - <40	72	20.22 %	59.27 %	16.25 %	24.85 %
40 - <50	65	18.26 %	77.53 %	14.46 %	22.75 %
50 - <60	54	15.17 %	92.70 %	11.69 %	19.42 %
60 - <65	8	2.25 %	94.94 %	1.05 %	4.55 %
65 - < HIVALUE	18	5.06 %	100.00 %	3.11 %	8.02 %
TOTAL	**356**	**100.00 %**	**100.00 %**		

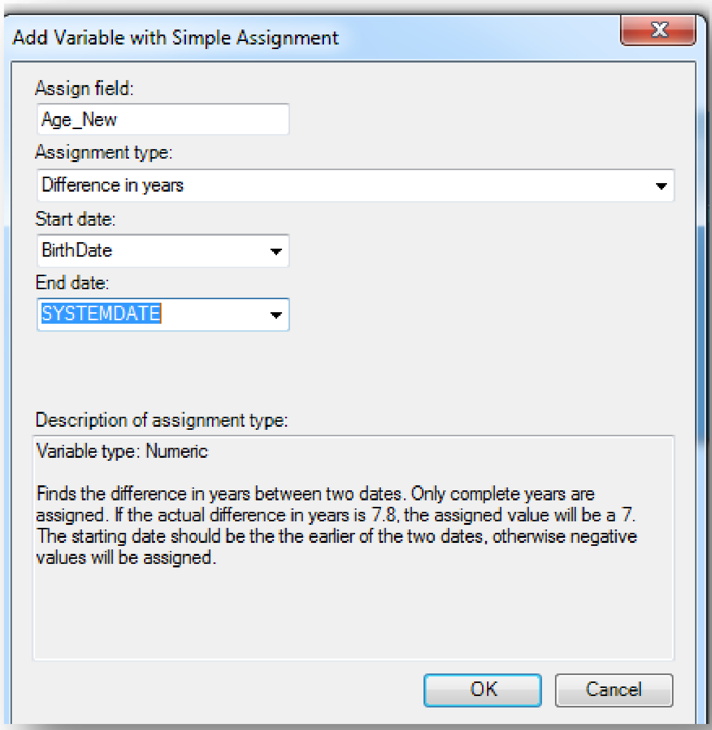
Figure 7.22: Age_Recoded Line List

The Age_Recoded variable contains eight distinct values, which makes analysis easier.

Simple Assignment

Simple Assignment will assign the value of an existing variable to a newly defined variable. The example below demonstrates Simple Assignment.

1. Select the **Sample.PRJ** Data Source.
2. From the Data Source Explorer menu, select **Surveillance**. Click **OK**.
3. On the left-hand side of the Visual Dashboard canvas, move the mouse cursor over the **Defined Variables** gadget. The gadget expands and becomes fully visible.
4. Click the **New Variable** button.
5. Select **With Simple Assignment**. The **Simple Assignment** dialog box appears.


Add Variable with Simple Assignment

Assign field: Age_New

Assignment type: Difference in years

Start date: BirthDate

End date: SYSTEMDATE

Description of assignment type:

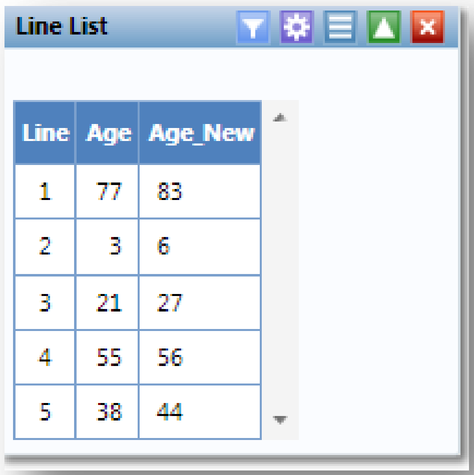
Variable type: Numeric

Finds the difference in years between two dates. Only complete years are assigned. If the actual difference in years is 7.8, the assigned value will be a 7. The starting date should be the the earlier of the two dates, otherwise negative values will be assigned.

Figure 7.23: Add Variable with Simple Assignment Options

6. In the Assign field, enter **Age_New**.
7. Select **Difference in years** from the **Assignment type** drop-down list.
8. Select **BirthDate** from the **Start Date** drop-down list.
9. Select **SYSTEMDATE** from the **End Date** drop-down list. The System Date is a system-level variable that represents the current date and time on your computer.
10. Click **OK**. A description of the assignment appears in the **Variable Definitions** textbox.

To see the **Age_New** variable in line list view, right click on the canvas and then select **Add Analysis Gadget > Line List**. Select **Age_New** from **List of Variables to display** and then click **Generate Line List.**



Line	Age	Age_New
1	77	83
2	3	6
3	21	27
4	55	56
5	38	44

Figure 7.24: Simple Assignment Line List

In this example, the current age of the patient is calculated from the difference between the birth date and today's date. The **Analysis Gadget > Line List** view displays the new variable created via Simple Assignment. For information regarding the Analysis Gadget, refer to the Displaying Statistics and Records section.

Conditional Assignment

Conditional Assignment defines conditions and one or more events that occur if the specified conditions are met. An alternative event can be given after the ELSE statement. The ELSE statement will be executed if the first set of conditions is not true. If the condition is true, the first event is executed. If the statement is false and no ELSE statement is selected, the statement is bypassed.

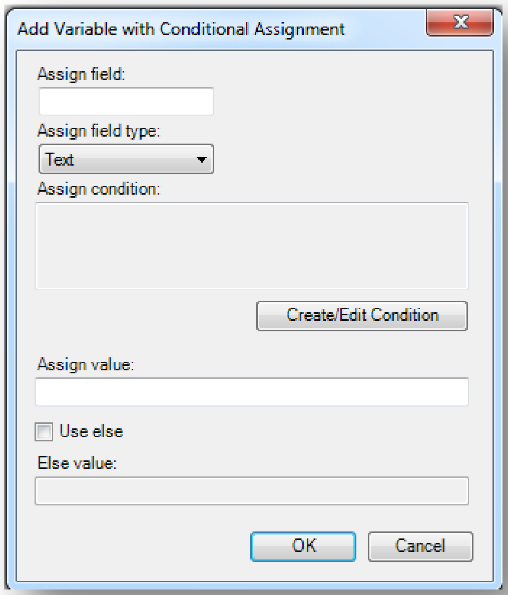
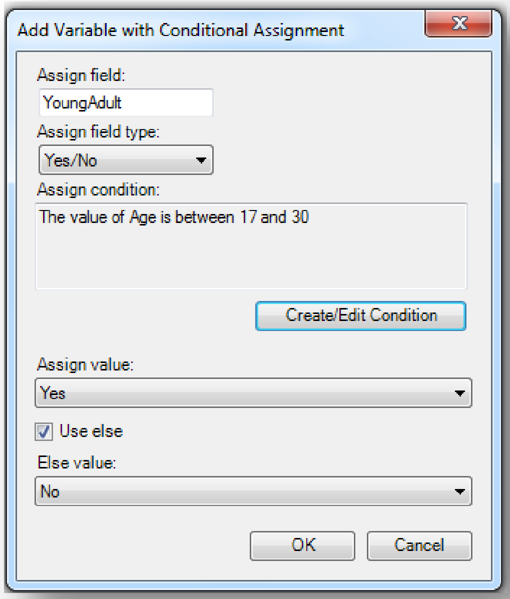


Figure 7.25: Add Variable with Conditional Assignment

- **Assign Field** indicates the new variable name.
- **Assign Field Type** indicates the format of the new variable. The available options are Text, Numeric, or Yes/No.
- **Assign Condition** determines whether subsequent commands will be run. If the condition specified in this textbox is true, the Assign Value will be applied.
- **Assign Value** states the value of the new variable when the Assign Condition is true.
- **Use Else** checkbox indicates whether an Else Value is specified. This checkbox is unmarked by default.
- The **Else Value** is applied when the statement in the Assign Condition textbox is false.

In the following example, the **YoungAdult** variable will be conditionally assigned the value of **Yes** if the Age variable has a value between (but not including) 17 and 30. If the Age value falls outside of this range, a value of **No** will be assigned to YoungAdult.

1. Select the **Sample.PRJ** Data Source.
2. From the **Data Source Explorer** menu, select **Surveillance**. Click **OK**.
3. On the left-hand side of the Visual Dashboard canvas, move the mouse cursor over the **Defined Variables** gadget. The gadget expands and becomes fully visible.
4. Click the **New Variable** button.
5. Select **With Conditional Assignment**. The **Add Variable with Conditional Assignment** dialog box appears.


Add Variable with Conditional Assignment

Assign field: YoungAdult

Assign field type: Yes/No

Assign condition: The value of Age is between 17 and 30

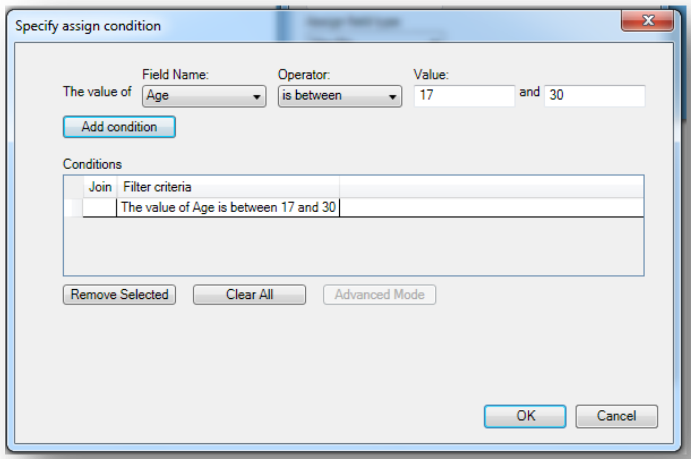
Assign value: Yes

Use else [✓]

Else value: No

Figure 7.26: Conditional Assignment/ Create Edit Condition

6. In the Assign field, enter **YoungAdult**.
7. Select **Yes/No** from the **Assign field type** drop-down list.
8. Click on the **Create/Edit Condition** button. The **Specify Assign Condition** dialog box appears.


Specify assign condition

The value of — Field Name: Age — Operator: is between — Value: 17 and 30

Conditions

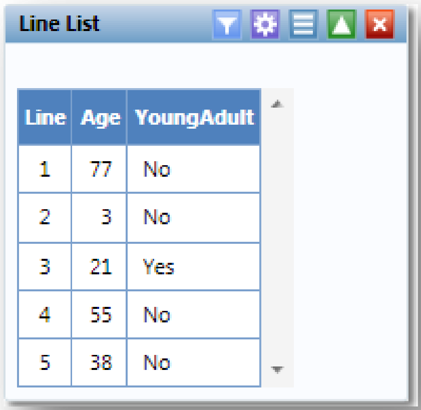
Join	Filter criteria
	The value of Age is between 17 and 30

Figure 7.27: Conditional Assignment Add Condition

9. Select **Age** from the **Field Name** drop-down list.
10. Select **is between** from the **Operator** drop-down list.
11. Enter **17** and **30** for Values. Click **Add Condition**.
12. Click **OK**. The **Add Variable with Conditional Assignment** dialog box appears.

13. The condition now appears in the **Assign Condition** textbox.
14. Select **Yes** from the **Assign Value** drop-down list.
15. Check the **Use Else** checkbox.
16. Select **No** from the **Else Value** drop-down list.
17. Click **OK**. A description of the assignment appears in the **Variable Definitions** textbox.

To see the **YoungAdult** variable in line list view, right click on the canvas and then select **Add Analysis Gadget > Line List**. Select **YoungAdult** from **List of Variables to display** and then click **Generate Line List.**



Line	Age	YoungAdult
1	77	No
2	3	No
3	21	Yes
4	55	No
5	38	No

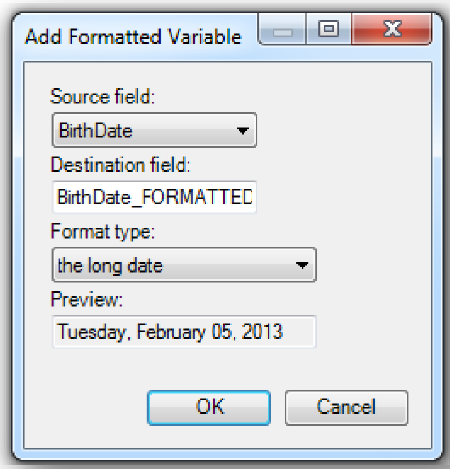
Figure 7.28: Conditional Assignment Line List

Note: The Else clause at the end of the IF statement is important to specify what should happen if the conditions are not met. Without the Else clause, YoungAdult would be null or missing for any values falling outside the 17 to 30 range. For information regarding the Analysis Gadget, refer to the Displaying Statistics and Records section.

Formatted Value

The format and name of an existing variable may be changed with the Formatted Value option. This is currently only useful with dates as demonstrated by the example below.

1. Select the **Sample.PRJ** Data Source.
2. From the **Data Source Explorer** menu, select **Surveillance**. Click **OK**.
3. On the left-hand side of the Visual Dashboard canvas, move the mouse cursor over the **Defined Variables** gadget. The gadget expands and becomes fully visible.
4. Click the **New Variable** button.
5. Select **With Formatted Value**. The **Add Formatted Value** dialog box appears.

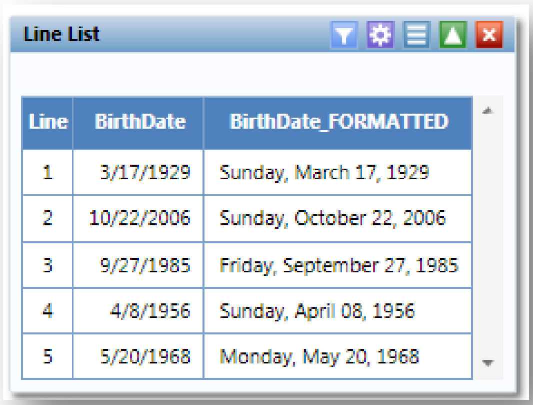


Source field: BirthDate
Destination field: BirthDate_FORMATTED
Format type: the long date
Preview: Tuesday, February 05, 2013

Figure 7.29: Add Formatted Variable Options

6. Select **BirthDate** from the **Source field** drop-down list.
7. The Destination field automatically shows the source field with a suffix of "FORMATTED". Modify the name of the Destination field as needed.
8. Select **the long date** from the **Format type** drop-down list. Using the current system date, a sample long date value appears in the Preview field. The current system date is used for the preview.
9. Click **OK**. A description of the assignment appears in the **Variable Definitions** textbox.

To see the **BirthDate_FORMATTED** variable in line list view, right click on the canvas and then select **Add Analysis Gadget > Line List**. Select **BirthDate_FORMATTED** from **List of Variables to display** and then click **Generate Line List.**



Line	BirthDate	BirthDate_FORMATTED
1	3/17/1929	Sunday, March 17, 1929
2	10/22/2006	Sunday, October 22, 2006
3	9/27/1985	Friday, September 27, 1985
4	4/8/1956	Sunday, April 08, 1956
5	5/20/1968	Monday, May 20, 1968

Figure 7.30: Formatted Variable Line List

The **Analysis Gadget > Line List** view displays the new variable created via Formatted Value. For information regarding the Analysis Gadget, refer to the Displaying Statistics and Records section.

Assigned Expression

Assigned Expression assigns the result of an expression or the field value to a variable as demonstrated in the example below. In the following example, the zip code field is a number. To use the zip code in a map, a new zip code variable must be defined and assigned text values.

1. Select the **Sample.PRJ** Data Source.
2. From the **Data Source Explorer** menu, select **Surveillance**. Click **OK**.
3. On the left-hand side of the Visual Dashboard canvas, move the mouse cursor over the **Defined Variables** gadget. The gadget expands and becomes fully visible.
4. Click the **New Variable** button.
5. Select **With Assigned Expression**. The **Add Variable with Expression** dialog box appears.

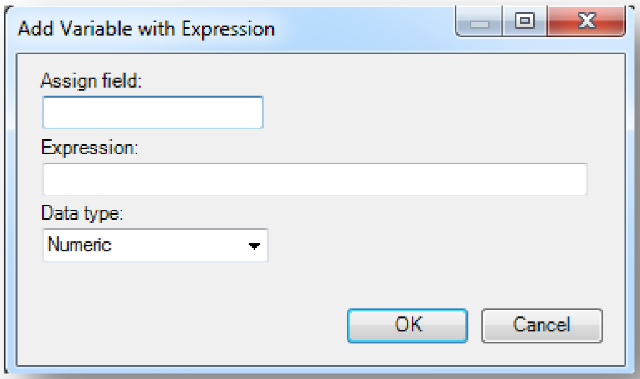
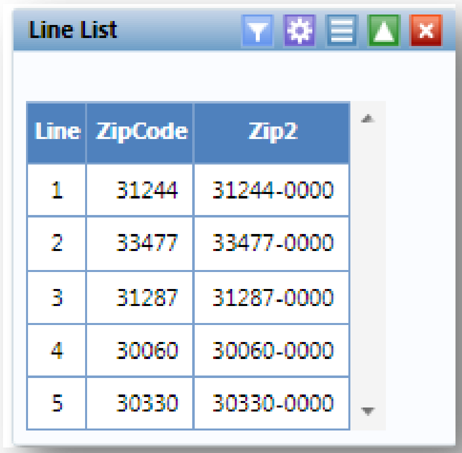


Figure 7.31: Add Variable with Expression

6. In the **Assign Field**, enter **Zip2**.
7. In the **Expression** field, type the syntax: **ZipCode + '-0000'**.
8. Select **Text** from the **Data type** drop-down list.
9. Click **OK**. The assignment appears in the Variable Definition textbox.

To see the **Zip2** variable in line list view, right click on the canvas and then select **Add Analysis Gadget > Line List**. Select **Zip2** from **List of Variables to display** and then click **Generate Line List.** For information regarding the Analysis Gadget, refer to the Displaying Statistics and Records section.



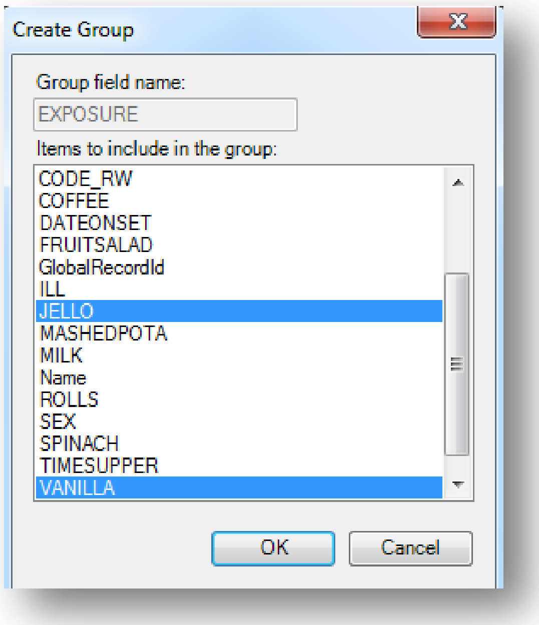
Line	ZipCode	Zip2
1	31244	31244-0000
2	33477	33477-0000
3	31287	31287-0000
4	30060	30060-0000
5	30330	30330-0000

Figure 7.32: Add Variable with Expression Line List

Creating a Variable Group

The Create variable group option in Visual Dashboard allows for the temporary grouping of variables. This option is useful when there are many variables and analyzing each is impractical. Group variables can be used in the Combined Frequency and MxN/2x2 gadgets. The example below demonstrates how to create a variable group.

1. Select the **Sample.PRJ** Data Source.
2. From the **Data Source Explorer** menu, select **Oswego**. Click **OK**.
3. On the left-hand side of the Visual Dashboard canvas, move the mouse cursor over the **Defined Variables** gadget. The gadget expands and becomes fully visible.
4. Click the **New Variable** button.
5. Select **Create Variable Group.** The **Create Group** dialog box appears.



EXPOSURE

CODE_RW
COFFEE
DATEONSET
FRUITSALAD
GlobalRecordId
ILL
JELLO
MASHEDPOTA
MILK
Name
ROLLS
SEX
SPINACH
TIMESUPPER
VANILLA

Figure 7.33: Create Group Variable Options

6. In the Group field name, enter **Exposure**.
7. Select **Jello** and **Vanilla** from the **Items to include in the group** list. Hold down the **Ctrl** key on your keyboard to select more than one variable at a time.
8. Click **OK**.
9. The variable group statement now appears in the **Variable Definitions** textbox.

Edit Variable

Edit Variable allows an existing variable definition to be changed. The following example will add a new variable to the Exposure group variable created above.

1. On the left-hand side of the Visual Dashboard canvas, move the mouse cursor over the **Defined Variables** gadget. The gadget expands and becomes fully visible.
2. Click the **Edit Variable** button.

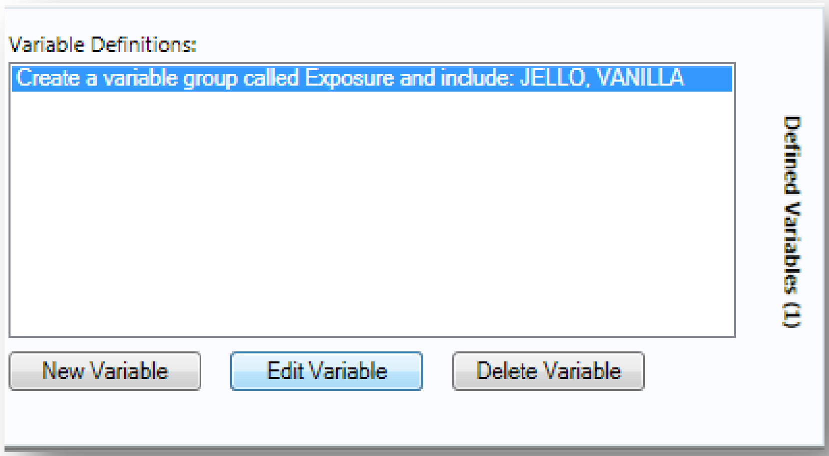


Figure 7.34: Defined Variable/Edit Variable

3. The **Create Group** dialog box appears.
4. Press and hold the **Ctrl** key on your keyboard and select **Spinach**.
5. Click **OK**.
6. The Exposure variable group now includes Spinach, which appears in the Variable Definitions text box.

Delete Variable

The Delete Variable option in the Data Recoding and Formatting Gadget removes a defined variable from the system. In the example below, the Exposure group variable will be deleted.

1. On the left-hand side of the Visual Dashboard canvas, move the mouse cursor over the **Defined Variables** gadget. The gadget expands and becomes fully visible.
2. Select the **Exposure** variable definition.

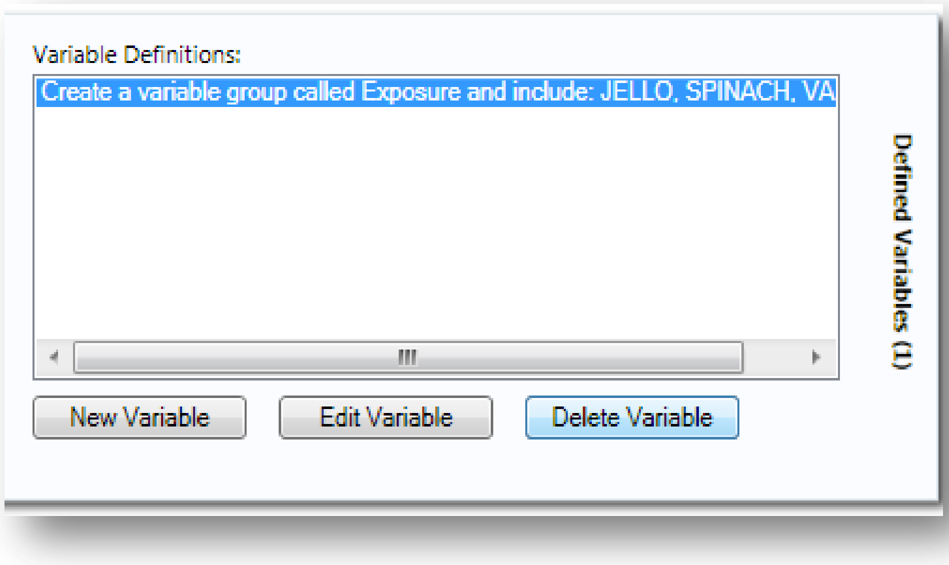


Figure 7.35: Defined Variable/Delete Variable

3. Click on the **Delete Variable** button.
4. The variable is deleted from the **Variable Definitions** text box.

Adding and Removing Data Filters

Add a Filter

Visual Dashboard typically opens all records in a file. The Data Filter Gadget in Visual Dashboard allows you to select a subset of data by specifying certain conditions. This is useful when trying to analyze a specific group of records. There are two modes in this gadget; guided and advanced. The Guided Mode allows you to apply conditions by selecting values from Field Name, Operator, and Value drop-down lists. Filters may be applied in the Advanced Mode by entering the filtering conditions in the free form text box. To add a filter, follow the steps below.

1. Select the **EColi.PRJ** Data Source.
2. From the **Data Source Explorer** menu, click **FoodHistory.**
3. Click **OK.**
4. From the right-hand side of the Visual Dashboard canvas, move the mouse over the **Data Filter** gadget. The gadget expands outwards.
5. From the **Field Name** drop-down list, select **Ill**.
6. From the **Operator** drop-down list, select **is equal to**.
7. From the **Value** drop-down list, select **Yes**.

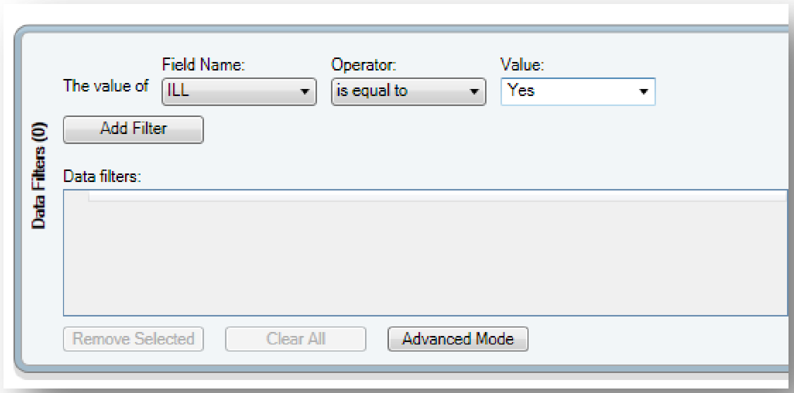
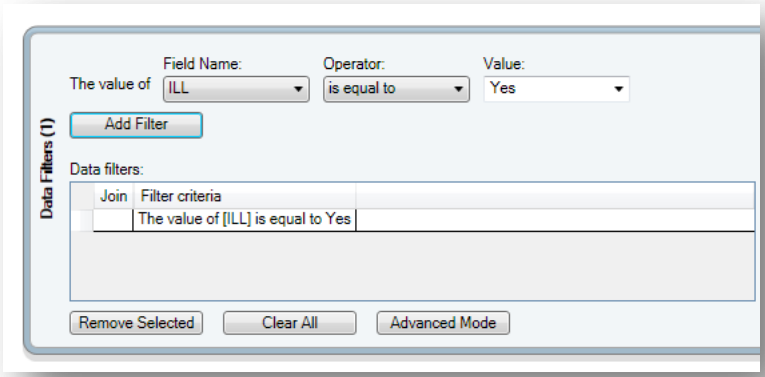


Figure 7.36: Data Filter Gadget/Add Filter

8. Click the **Add Filter** button. The filter condition is added to the Data filters grid view.



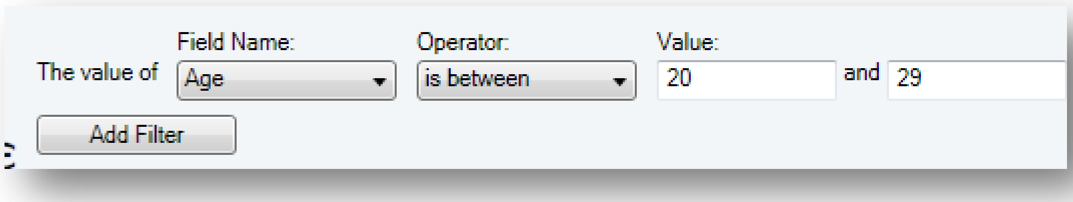
Join	Filter criteria
	The value of [ILL] is equal to Yes

Figure 7.37: Data Filter Gadget/Add Filter with Condition

If a filter condition is added or removed, all gadgets on the canvas will automatically refresh. The record count at the top now reads **276**, indicating that there are 276 out of 359 records where the value of ILL is Yes.

Follow the steps below to add an additional filter:

1. On the right-hand side of the Visual Dashboard canvas, move the mouse over the **Data Filter** gadget. The gadget expands outwards.
2. From the **Field Name** drop-down list, select **Age**.
3. From the **Operator** drop-down list, select **is between**.
4. From the **Value** text boxes, type **20** for the first box.
5. From the **Value** text boxes, type **29** for the second box.

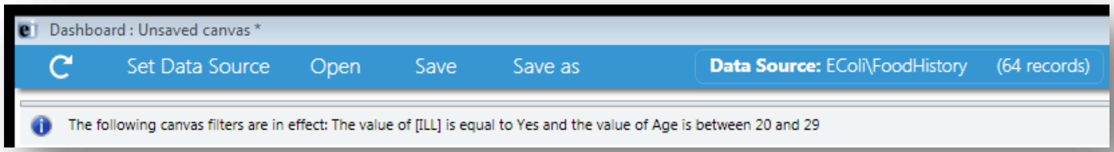


The value of Age is between 20 and 29

Figure 7.38: Data Filter Gadget/Add Filter/Specify Condition

6. Click **Add Filter**. A context menu appears asking if you want to add this condition using an AND or an OR.
7. Select the **AND** option. The filter condition is added to the Data filters grid view beneath the first condition.

The record count shows 64 records. Only 64 records match the two established filter criterion. The patient must have been ill and be between 20 and 29 years of age. All gadgets added to the Visual Dashboard canvas will only display data from these 64 records. The filter settings appear at the top of the canvas.



Data Source: EColi\FoodHistory (64 records)

The following canvas filters are in effect: The value of [ILL] is equal to Yes and the value of Age is between 20 and 29

Figure 7.39: Data Filter Setting

Remove Selected Filter

The data filters must be cleared to continue working with the full dataset. Follow the steps below to remove a filter.

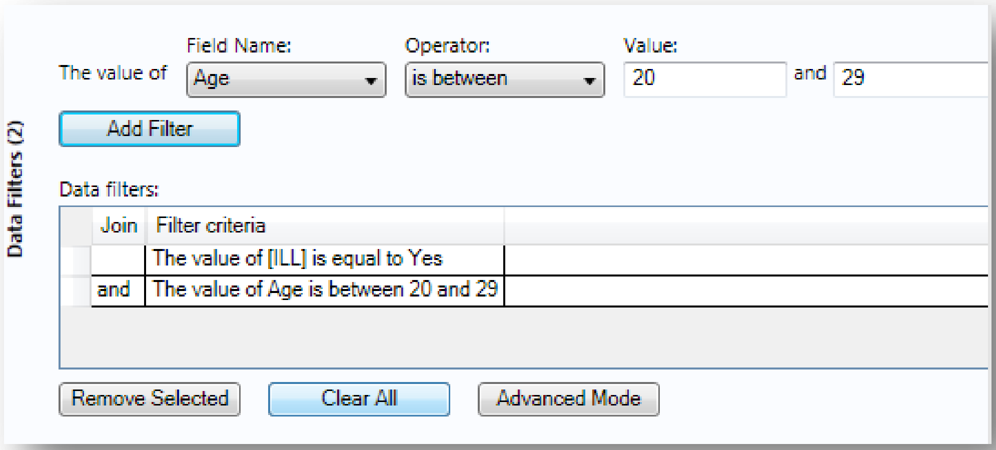
1. On the right-hand side of the Visual Dashboard canvas, move the mouse over the **Data Filter** gadget. The gadget expands outwards.
2. Select the **second condition**, which should appear as "**The value of Age is between 20 and 29**."
3. Click the **Remove Selected** button. The condition disappears.
4. Select the only remaining **condition**.
5. Click the **Remove Selected** button. The condition disappears.

Clear All Filters

The **Clear All** button removes all existing filters at once as demonstrated in the example below.

1. On the right-hand side of the Visual Dashboard canvas, move the mouse over the **Data Filter** gadget. The gadget expands outwards.

2. Click the **Clear All** button. All existing filter conditions disappear.



Join	Filter criteria
	The value of [ILL] is equal to Yes
and	The value of Age is between 20 and 29

Figure 7.40: Data Filter Gadget/Add Filter with Two Conditions

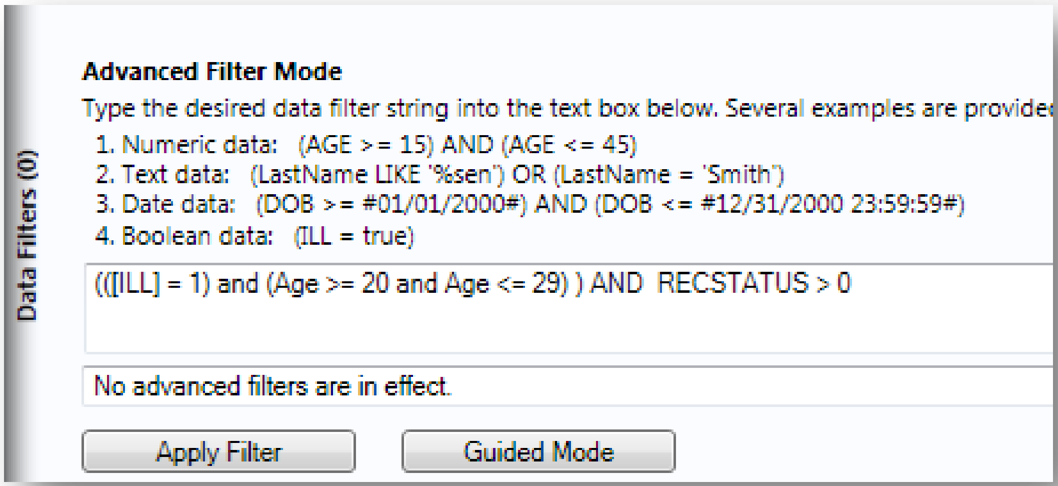
Advanced Filter Mode

The advanced filter mode allows you to type the desired data filter string into the text box provided. Once the filter is applied, the canvas will only display the records that meet the specified condition(s). The Data Filtering Gadget is located on the right-hand side of the Visual Dashboard canvas. The gadget expands outwards.

Note: Advanced Filtering uses the .NET DataView.RowFilter syntax. For more information about this syntax, refer to Microsoft's website describing the DataView.RowFilter Property.

The example below demonstrates how to use advanced filters. Select **Advanced Mode.** The Advanced Filter Mode appears.

1. In the text box, enter the following condition **"(([ILL] = 1) and (Age >= 20 and Age <= 29)) AND RECSTATUS > 0".**
2. Click the **Apply Filter** button.



Advanced Filter Mode

Type the desired data filter string into the text box below. Several examples are provide

1. Numeric data: (AGE >= 15) AND (AGE <= 45)
2. Text data: (LastName LIKE '%sen') OR (LastName = 'Smith')
3. Date data: (DOB >= #01/01/2000#) AND (DOB <= #12/31/2000 23:59:59#)
4. Boolean data: (ILL = true)

```
(([ILL] = 1) and (Age >= 20 and Age <= 29) ) AND  RECSTATUS > 0
```

No advanced filters are in effect.

Figure 7.41: Advanced Filter Mode

3. The verbiage "**No advanced filters are in effect**" will be replaced with "**The filter is valid and is now in effect.**" When filters are applied, the conditions appear in green font.
4. The record count in the top-left corner of the Visual Dashboard canvas now reads **64**, indicating that there are 64 out of 359 records where the value of ILL is Yes and the value for age is between 20 and 29.

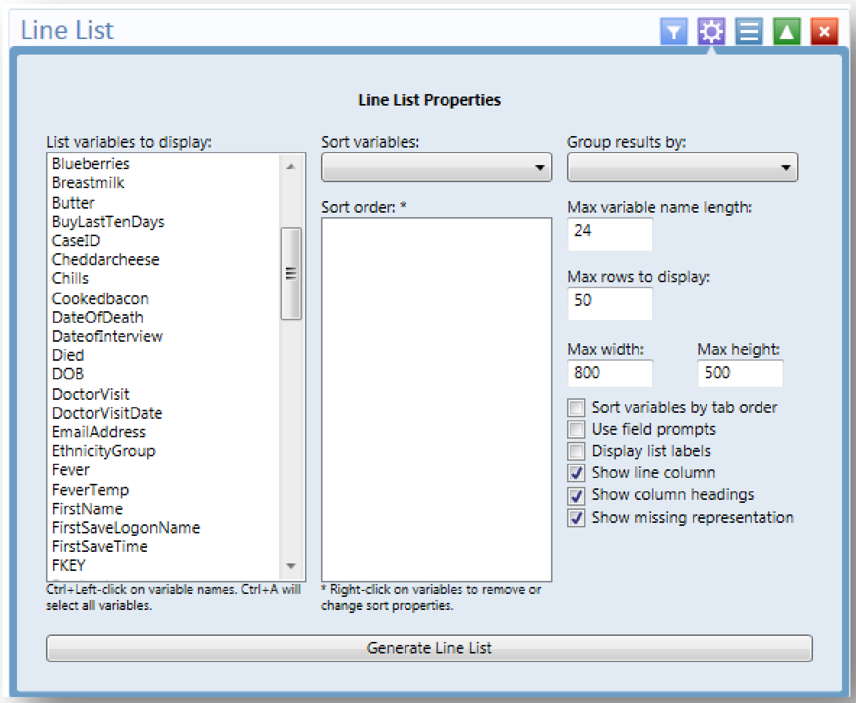
Note: You can toggle between the Guided Mode and the Advanced Mode by clicking on the respective button on each screen.

Displaying Statistics and Records

Analysis Gadget > Line List

The Line List option creates a list of the current dataset. This allows you to view the details of the data in your current data source. You can select variables to display and then sort or group those variables accordingly. Line List supports adding groups or pages. All fields in a group or on a page will be displayed in such cases. Groups and pages always appear at the bottom of the list.

1. Select the **EColi.PRJ** Data Source. Open the **FoodHistory** form from the **Data Source Explorer** menu.
2. Right click on the canvas and select **Add Analysis Gadget** > **Line list**.
3. The Line List gadget appears on the Visual Dashboard canvas.



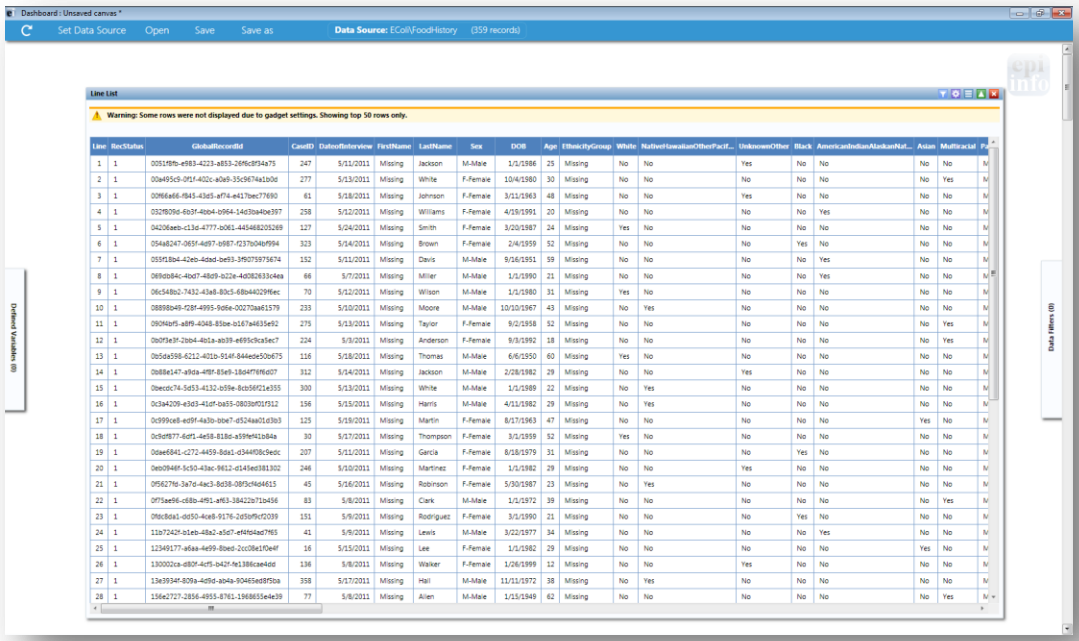
Max variable name length: 24
Max rows to display: 50
Max width: 800
Max height: 500
☐ Sort variables by tab order
☐ Use field prompts
☐ Display list labels
☑ Show line column
☑ Show column headings
☑ Show missing representation

Ctrl+Left-click on variable names. Ctrl+A will select all variables.

* Right-click on variables to remove or change sort properties.

Figure 7.42: Line List Properties

4. To select multiple variables from **List variables to display,** hold down the **Ctrl** key and click on the variables to display. The **Ctrl + A** combination selects all variables at once.
5. Click the **Generate Line List** button. The line list appears on the canvas.



Data Source: EColi\FoodHistory (359 records)

Warning: Some rows were not displayed due to gadget settings. Showing top 50 rows only.

Line	RecStatus	GlobalRecordId	CaseID	DateofInterview	FirstName	LastName	Sex	DOB	Age	EthnicityGroup	White	NativeHawaiianOtherPacif...	UnknownOther	Black	AmericanIndianAlaskanNat...	Asian	Multiracial	Pa
1	1	0051f8fb-e983-4223-a853-26f6c8f34a75	247	5/11/2011	Missing	Jackson	M-Male	1/1/1986	25	Missing	No	No	Yes	No	No	No	No	M
2	1	00a495c9-0f1f-402c-a0a9-35c9674a1b0d	277	5/13/2011	Missing	White	F-Female	10/4/1980	30	Missing	No	No	No	No	No	No	Yes	M
3	1	00f66a66-f845-43d5-af74-e417bec77690	61	5/18/2011	Missing	Johnson	F-Female	3/11/1963	48	Missing	No	No	Yes	No	No	No	No	M
4	1	032f809d-6b3f-4bb4-b964-14d3ba4be397	258	5/12/2011	Missing	Williams	F-Female	4/19/1991	20	Missing	No	No	No	No	Yes	No	No	M
5	1	04206aeb-c13d-4777-b061-445468205269	127	5/24/2011	Missing	Smith	F-Female	3/20/1987	24	Missing	Yes	No	No	No	No	No	No	M
6	1	054a8247-065f-4d97-b987-f237b04bf994	323	5/14/2011	Missing	Brown	F-Female	2/4/1959	52	Missing	No	No	No	Yes	No	No	No	M
7	1	055f18b4-42eb-4dad-be93-3f9075975674	152	5/11/2011	Missing	Davis	M-Male	9/16/1951	59	Missing	No	No	No	No	Yes	No	No	M
8	1	069db84c-4bd7-48d9-b22e-4d082633c4ea	66	5/7/2011	Missing	Miller	M-Male	1/1/1990	21	Missing	No	No	No	No	Yes	No	No	M
9	1	06c548b2-7432-43a8-80c5-68b44029f6ec	70	5/12/2011	Missing	Wilson	M-Male	1/1/1980	31	Missing	Yes	No	No	No	No	No	No	M
10	1	08898b49-f28f-4995-9d6e-00270aa61579	233	5/10/2011	Missing	Moore	M-Male	10/10/1967	43	Missing	No	Yes	No	No	No	No	No	M
11	1	090f4bf5-a8f9-4048-85be-b167a4635e92	275	5/13/2011	Missing	Taylor	F-Female	9/2/1958	52	Missing	No	No	No	No	No	No	Yes	M
12	1	0b0f3e3f-2bb4-4b1a-ab39-e695c9ca5ec7	224	5/3/2011	Missing	Anderson	F-Female	9/3/1992	18	Missing	No	No	No	No	No	No	Yes	M
13	1	0b5da598-6212-401b-914f-844ede50b675	116	5/18/2011	Missing	Thomas	M-Male	6/6/1950	60	Missing	Yes	No	No	No	No	No	No	M
14	1	0b88e147-a9da-4f8f-85e9-18d4f76f6d07	312	5/14/2011	Missing	Jackson	M-Male	2/28/1982	29	Missing	No	No	Yes	No	No	No	No	M
15	1	0becdc74-5d53-4132-b59e-8cb56f21e355	300	5/13/2011	Missing	White	M-Male	1/1/1989	22	Missing	No	Yes	No	No	No	No	No	M
16	1	0c3a4209-e3d3-41df-ba55-0803bf01f312	156	5/15/2011	Missing	Harris	M-Male	4/11/1982	29	Missing	No	Yes	No	No	No	No	No	M
17	1	0c999ce8-ed9f-4a3b-bbe7-d524aa01d3b3	125	5/19/2011	Missing	Martin	F-Female	8/17/1963	47	Missing	No	No	No	No	No	Yes	No	M
18	1	0c9df877-6df1-4e58-818d-a59fef41b84a	30	5/17/2011	Missing	Thompson	F-Female	3/1/1959	52	Missing	Yes	No	No	No	No	No	No	M
19	1	0dae6841-c272-4459-8da1-d344f08c9edc	207	5/11/2011	Missing	Garcia	F-Female	8/18/1979	31	Missing	No	No	No	Yes	No	No	No	M
20	1	0eb0946f-5c50-43ac-9612-d145ed381302	246	5/10/2011	Missing	Martinez	F-Female	1/1/1982	29	Missing	No	No	Yes	No	No	No	No	M
21	1	0f5627fd-3a7d-4ac3-8d38-08f3cf4d4615	45	5/16/2011	Missing	Robinson	F-Female	5/30/1987	23	Missing	No	Yes	No	No	No	No	No	M
22	1	0f75ae96-c68b-4f91-af63-38422b71b456	83	5/8/2011	Missing	Clark	M-Male	1/1/1972	39	Missing	No	No	No	No	No	No	Yes	M
23	1	0fdc8da1-dd50-4ce8-9176-2d5bf9cf2039	151	5/9/2011	Missing	Rodriguez	F-Female	3/1/1990	21	Missing	No	No	No	Yes	No	No	No	M
24	1	11b7242f-b1eb-48a2-a5d7-ef4fd4ad7f65	41	5/9/2011	Missing	Lewis	M-Male	3/22/1977	34	Missing	No	No	No	No	Yes	No	No	M
25	1	12349177-a6aa-4e99-8bed-2cc08e1f0e4f	16	5/15/2011	Missing	Lee	F-Female	1/1/1982	29	Missing	No	No	No	No	No	Yes	No	M
26	1	130002ca-d80f-4cf5-b42f-fe1386cae4dd	136	5/8/2011	Missing	Walker	F-Female	1/26/1999	12	Missing	No	No	Yes	No	No	No	No	M
27	1	13e3934f-809a-4d9d-ab4a-90465ed8f5ba	358	5/17/2011	Missing	Hall	M-Male	11/11/1972	38	Missing	No	Yes	No	No	No	No	No	M
28	1	156e2727-2856-4955-8761-1968655e4e39	77	5/8/2011	Missing	Allen	M-Male	1/15/1949	62	Missing	No	No	No	No	No	No	Yes	M

Figure 7.43: Generate Line List

Note: There is a yellow warning at the top of the output. By default, the line list will only show the top 50 rows. The row limit exists to improve performance when working with large datasets. The Max Rows to Display setting can be modified in the Line List Properties panel.

Sorting Records

Sometimes it is important to have the records in a file arranged in a particular order, for example alphabetically by name, or numerically by age. There are two ways to sort the records; by clicking on the column name in the line list or by using the sort options in the line list properties dialog box.

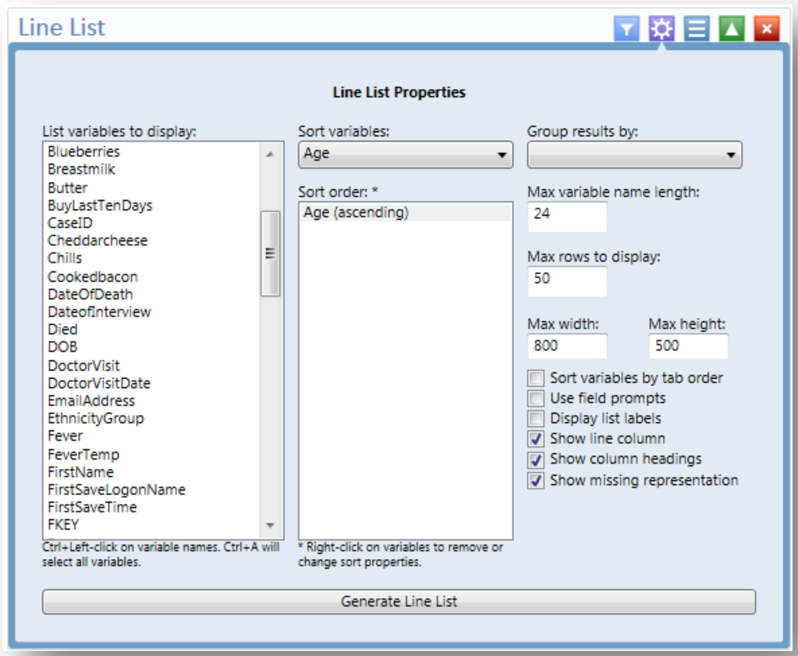
Sorting a List Using by Column Name

Follow the steps below to sort the list by Age.

1. Select the **EColi.PRJ** Data Source. Open the **FoodHistory** form from the **Data Source Explorer** menu.
2. Right click on the canvas and select **Add Analysis Gadget** > **Line list**.
3. Select all variables from **List Variables to Display** box by pressing **Ctrl + A** on your keyboard.
4. Click **Generate Line List**.
5. The **Line List** gadget appears on the Visual Dashboard canvas.
6. Click on the **Age** column heading**.**
7. The records are displayed by **Age** in ascending order.

Sorting a List Using Line List Properties

1. Select the **EColi.PRJ** Data Source. Open the **FoodHistory** form from the **Data Source Explorer** menu.
2. Right click on the canvas and select **Add Analysis Gadget** > **Line list**.
3. The **Line List** gadget appears on the Visual Dashboard canvas.
4. Select all variables from **List Variables to Display** box by pressing **Ctrl + A** on your keyboard.
5. The list data is not sorted in any particular order. To sort variables, select the variable from the **Sort variables** drop-down list. The selected variable will appear in the **Sort Order** box.


Line List

Line List Properties

List variables to display: Blueberries, Breastmilk, Butter, BuyLastTenDays, CaseID, Cheddarcheese, Chills, Cookedbacon, DateOfDeath, DateofInterview, Died, DOB, DoctorVisit, DoctorVisitDate, EmailAddress, EthnicityGroup, Fever, FeverTemp, FirstName, FirstSaveLogonName, FirstSaveTime, FKEY

Ctrl+Left-click on variable names. Ctrl+A will select all variables.

Sort variables: Age

Sort order: * Age (ascending)

* Right-click on variables to remove or change sort properties.

Group results by:

Max variable name length: 24

Max rows to display: 50

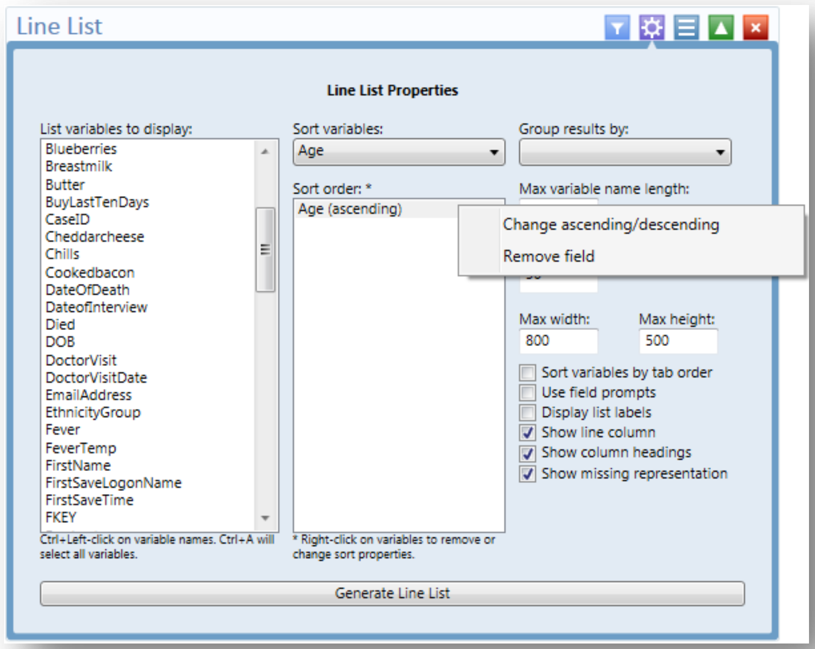
Max width: 800

Max height: 500

☐ Sort variables by tab order
☐ Use field prompts
☐ Display list labels
☑ Show line column
☑ Show column headings
☑ Show missing representation

Figure 7.44: Line List/ Sort Order

6. Right click on the variable in the **Sort Order** box to remove the sort variable or to change the sort order from ascending to descending.



Line List

Line List Properties

List variables to display:

Blueberries
Breastmilk
Butter
BuyLastTenDays
CaseID
Cheddarcheese
Chills
Cookedbacon
DateOfDeath
DateofInterview
Died
DOB
DoctorVisit
DoctorVisitDate
EmailAddress
EthnicityGroup
Fever
FeverTemp
FirstName
FirstSaveLogonName
FirstSaveTime
FKEY

Ctrl+Left-click on variable names. Ctrl+A will select all variables.

Sort variables: Age

Sort order: *

Age (ascending)

* Right-click on variables to remove or change sort properties.

Change ascending/descending
Remove field

Group results by:

Max variable name length:

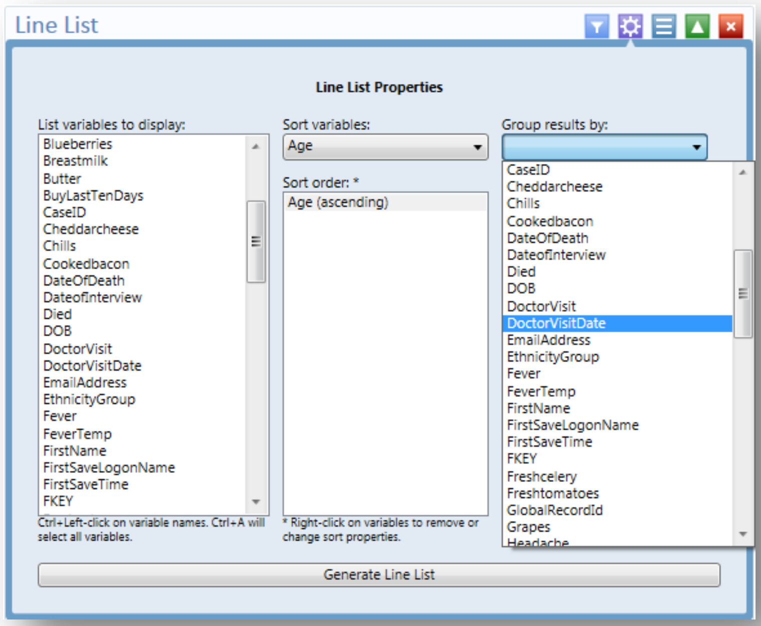
Max width: 800
Max height: 500

- [ ] Sort variables by tab order
- [ ] Use field prompts
- [ ] Display list labels
- [x] Show line column
- [x] Show column headings
- [x] Show missing representation

Generate Line List

Figure 7.45: Line List/Sort Order options

7. Select a variable from the **Group results by** drop-down list to group the list by a particular field.



Line List

Line List Properties

List variables to display:

Blueberries
Breastmilk
Butter
BuyLastTenDays
CaseID
Cheddarcheese
Chills
Cookedbacon
DateOfDeath
DateofInterview
Died
DOB
DoctorVisit
DoctorVisitDate
EmailAddress
EthnicityGroup
Fever
FeverTemp
FirstName
FirstSaveLogonName
FirstSaveTime
FKEY

Ctrl+Left-click on variable names. Ctrl+A will select all variables.

Sort variables: Age

Sort order: *

Age (ascending)

* Right-click on variables to remove or change sort properties.

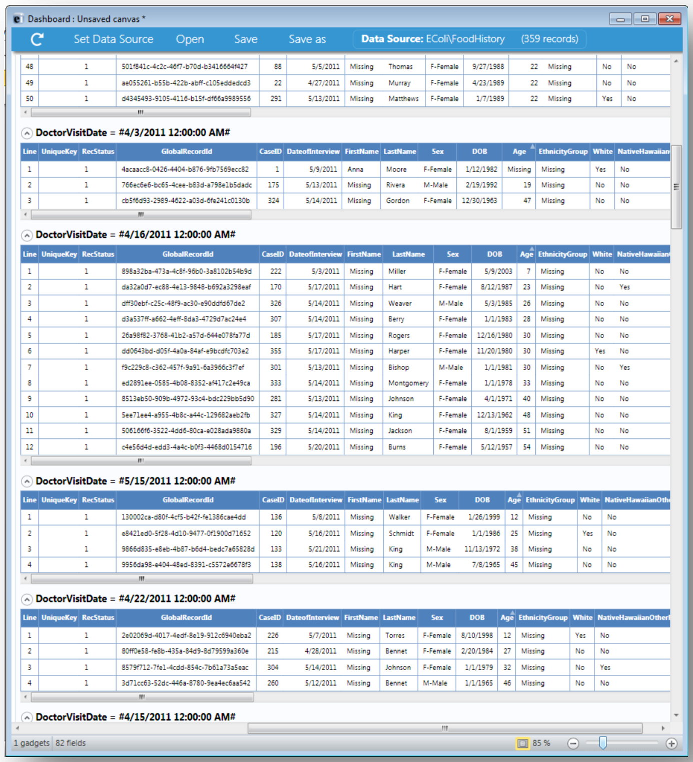
Group results by:

CaseID
Cheddarcheese
Chills
Cookedbacon
DateOfDeath
DateofInterview
Died
DOB
DoctorVisit
DoctorVisitDate
EmailAddress
EthnicityGroup
Fever
FeverTemp
FirstName
FirstSaveLogonName
FirstSaveTime
FKEY
Freshcelery
Freshtomatoes
GlobalRecordId
Grapes
Headache

Generate Line List

Figure 7.46: Line List/Group results options

8. Click the **Generate Line List** button. A separate line list, grouped by DoctorVisitDate, is generated and displayed. The records in each list are ordered by **Age**.


Dashboard : Unsaved canvas *
Set Data Source
Open
Save
Save as
Data Source: EColi\FoodHistory (359 records)

48		1	501f841c-4c2c-46f7-b70d-b3416664f427	88	5/5/2011	Missing	Thomas	F-Female	9/27/1988	22	Missing	No	No
49		1	ae055261-b55b-422b-abff-c105eddedcd3	22	4/27/2011	Missing	Murray	F-Female	4/23/1989	22	Missing	No	No
50		1	d4345493-9105-4116-b15f-df66a9989556	291	5/13/2011	Missing	Matthews	F-Female	1/7/1989	22	Missing	Yes	No

DoctorVisitDate = #4/3/2011 12:00:00 AM#

Line	UniqueKey	RecStatus	GlobalRecordId	CaseID	DateofInterview	FirstName	LastName	Sex	DOB	Age	EthnicityGroup	White	NativeHawaiian
1		1	4acaacc8-0426-4404-b876-9fb7569ecc82	1	5/9/2011	Anna	Moore	F-Female	1/12/1982	Missing	Missing	Yes	No
2		1	766ec6e6-bc65-4cee-b83d-a798e1b5dadc	175	5/13/2011	Missing	Rivera	M-Male	2/19/1992	19	Missing	No	No
3		1	cb5f6d93-2989-4622-a03d-6fe241c0130b	324	5/14/2011	Missing	Gordon	F-Female	12/30/1963	47	Missing	No	No

DoctorVisitDate = #4/16/2011 12:00:00 AM#

Line	UniqueKey	RecStatus	GlobalRecordId	CaseID	DateofInterview	FirstName	LastName	Sex	DOB	Age	EthnicityGroup	White	NativeHawaiian
1		1	898a32ba-473a-4c8f-96b0-3a8102b54b9d	222	5/3/2011	Missing	Miller	F-Female	5/9/2003	7	Missing	No	No
2		1	da32a0d7-ec88-4e13-9848-b692a3298eaf	170	5/17/2011	Missing	Hart	F-Female	8/12/1987	23	Missing	No	Yes
3		1	dff30ebf-c25c-48f9-ac30-e90ddfd67de2	326	5/14/2011	Missing	Weaver	M-Male	5/3/1985	26	Missing	No	No
4		1	d3a537ff-a662-4eff-8da3-4729d7ac24e4	307	5/14/2011	Missing	Berry	F-Female	1/1/1983	28	Missing	No	No
5		1	26a98f82-3768-41b2-a57d-644e078fa77d	185	5/17/2011	Missing	Rogers	F-Female	12/16/1980	30	Missing	No	No
6		1	dd0643bd-d05f-4a0a-84af-e9bcdfc703e2	355	5/17/2011	Missing	Harper	F-Female	11/20/1980	30	Missing	Yes	No
7		1	f9c229c8-c362-457f-9a91-6a3966c3f7ef	301	5/13/2011	Missing	Bishop	M-Male	1/1/1981	30	Missing	No	Yes
8		1	ed2891ee-0585-4b08-8352-af417c2e49ca	333	5/14/2011	Missing	Montgomery	F-Female	1/1/1978	33	Missing	No	No
9		1	8513eb50-909b-4972-93c4-bdc229bb5d90	281	5/13/2011	Missing	Johnson	F-Female	4/1/1971	40	Missing	No	No
10		1	5ee71ee4-a955-4b8c-a44c-129682aeb2fb	327	5/14/2011	Missing	King	F-Female	12/13/1962	48	Missing	No	No
11		1	506166f6-3522-4dd6-80ca-e028ada9880a	329	5/14/2011	Missing	Jackson	F-Female	8/1/1959	51	Missing	No	No
12		1	c4e56d4d-edd3-4a4c-b0f3-4468d0154716	196	5/20/2011	Missing	Burns	F-Female	5/12/1957	54	Missing	No	No

DoctorVisitDate = #5/15/2011 12:00:00 AM#

Line	UniqueKey	RecStatus	GlobalRecordId	CaseID	DateofInterview	FirstName	LastName	Sex	DOB	Age	EthnicityGroup	White	NativeHawaiianOth
1		1	130002ca-d80f-4cf5-b42f-fe1386cae4dd	136	5/8/2011	Missing	Walker	F-Female	1/26/1999	12	Missing	No	No
2		1	e8421ed0-5f28-4d10-9477-0f1900d71652	120	5/16/2011	Missing	Schmidt	F-Female	1/1/1986	25	Missing	Yes	No
3		1	9866d835-e8eb-4b87-b6d4-bedc7a65828d	133	5/21/2011	Missing	King	M-Male	11/13/1972	38	Missing	No	No
4		1	9956da98-e404-48ed-8391-c5572e6678f3	138	5/16/2011	Missing	King	M-Male	7/8/1965	45	Missing	No	No

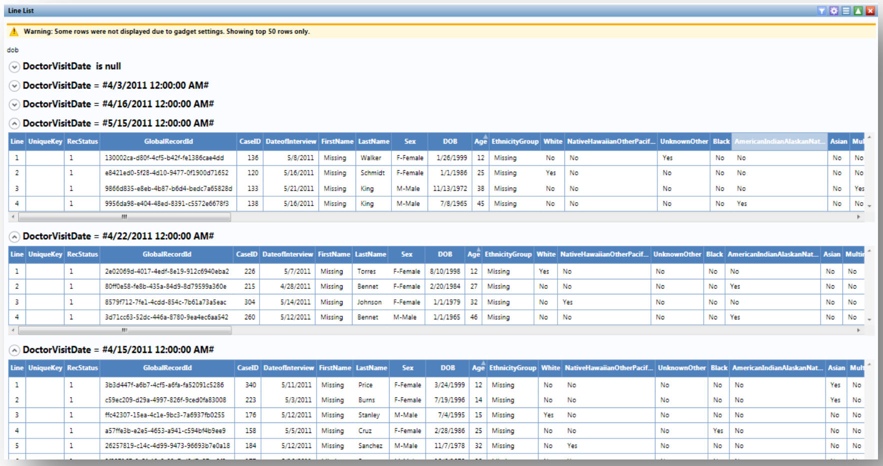
DoctorVisitDate = #4/22/2011 12:00:00 AM#

Line	UniqueKey	RecStatus	GlobalRecordId	CaseID	DateofInterview	FirstName	LastName	Sex	DOB	Age	EthnicityGroup	White	NativeHawaiianOther
1		1	2e02069d-4017-4edf-8e19-912c6940eba2	226	5/7/2011	Missing	Torres	F-Female	8/10/1998	12	Missing	Yes	No
2		1	80ff0e58-fe8b-435a-84d9-8d79599a360e	215	4/28/2011	Missing	Bennet	F-Female	2/20/1984	27	Missing	No	No
3		1	8579f712-7fe1-4cdd-854c-7b61a73a5eac	304	5/14/2011	Missing	Johnson	F-Female	1/1/1979	32	Missing	No	Yes
4		1	3d71cc63-52dc-446a-8780-9ea4ec6aa542	260	5/12/2011	Missing	Bennet	M-Male	1/1/1965	46	Missing	No	No

DoctorVisitDate = #4/15/2011 12:00:00 AM#

Figure 7.47: Line List with Group results

9. A list may be hidden from view by clicking on the arrow to the left of the list heading.



Line List

Warning: Some rows were not displayed due to gadget settings. Showing top 50 rows only.

dob

DoctorVisitDate is null

DoctorVisitDate = #4/3/2011 12:00:00 AM#

DoctorVisitDate = #4/16/2011 12:00:00 AM#

DoctorVisitDate = #5/15/2011 12:00:00 AM#

Line	UniqueKey	RecStatus	GlobalRecordId	CaseID	DateofInterview	FirstName	LastName	Sex	DOB	Age	EthnicityGroup	White	NativeHawaiianOtherPacif...	UnknownOther	Black	AmericanIndianAlaskanNat...	Asian	Mul
1		1	130002ca-d80f-4cf5-b42f-fe1386cae4dd	136	5/8/2011	Missing	Walker	F-Female	1/26/1999	12	Missing	No	No	Yes	No	No	No	No
2		1	e8421ed0-5f28-4d10-9477-0f1900d71652	120	5/16/2011	Missing	Schmidt	F-Female	1/1/1986	25	Missing	Yes	No	No	No	No	No	No
3		1	9866d835-e8eb-4b87-b6d4-bedc7a65828d	133	5/21/2011	Missing	King	M-Male	11/13/1972	38	Missing	No	No	No	No	No	No	Yes
4		1	9956da98-e404-48ed-8391-c5572e6678f3	138	5/16/2011	Missing	King	M-Male	7/8/1965	45	Missing	No	No	No	No	Yes	No	No

DoctorVisitDate = #4/22/2011 12:00:00 AM#

Line	UniqueKey	RecStatus	GlobalRecordId	CaseID	DateofInterview	FirstName	LastName	Sex	DOB	Age	EthnicityGroup	White	NativeHawaiianOtherPacif...	UnknownOther	Black	AmericanIndianAlaskanNat...	Asian	Multi
1		1	2e02069d-4017-4edf-8e19-912c6940eba2	226	5/7/2011	Missing	Torres	F-Female	8/10/1998	12	Missing	Yes	No	No	No	No	No	No
2		1	80ff0e58-fe8b-435a-84d9-8d79599a360e	215	4/28/2011	Missing	Bennet	F-Female	2/20/1984	27	Missing	No	No	No	No	Yes	No	No
3		1	8579f712-7fe1-4cdd-854c-7b61a73a5eac	304	5/14/2011	Missing	Johnson	F-Female	1/1/1979	32	Missing	No	Yes	No	No	No	No	No
4		1	3d71cc63-52dc-446a-8780-9ea4ec6aa542	260	5/12/2011	Missing	Bennet	M-Male	1/1/1965	46	Missing	No	No	No	No	Yes	No	No

DoctorVisitDate = #4/15/2011 12:00:00 AM#

Line	UniqueKey	RecStatus	GlobalRecordId	CaseID	DateofInterview	FirstName	LastName	Sex	DOB	Age	EthnicityGroup	White	NativeHawaiianOtherPacif...	UnknownOther	Black	AmericanIndianAlaskanNat...	Asian	Mult
1		1	3b3d447f-a6b7-4cf5-a6fa-fa52091c5286	340	5/11/2011	Missing	Price	F-Female	3/24/1999	12	Missing	No	No	No	No	No	Yes	No
2		1	c59ec209-d29a-4997-826f-9ced0fa83008	223	5/3/2011	Missing	Burns	F-Female	7/19/1996	14	Missing	No	No	No	No	No	Yes	No
3		1	ffc42307-15ea-4c1e-9bc3-7a6937fb0255	176	5/12/2011	Missing	Stanley	M-Male	7/4/1995	15	Missing	Yes	No	No	No	No	No	No
4		1	a57ffe3b-e2e5-4653-a941-c594bf4b9ee9	158	5/5/2011	Missing	Cruz	F-Female	2/28/1986	25	Missing	No	No	No	Yes	No	No	No
5		1	26257819-c14c-4d99-9473-96693b7e0a18	184	5/12/2011	Missing	Sanchez	M-Male	11/7/1978	32	Missing	No	Yes	No	No	No	No	No

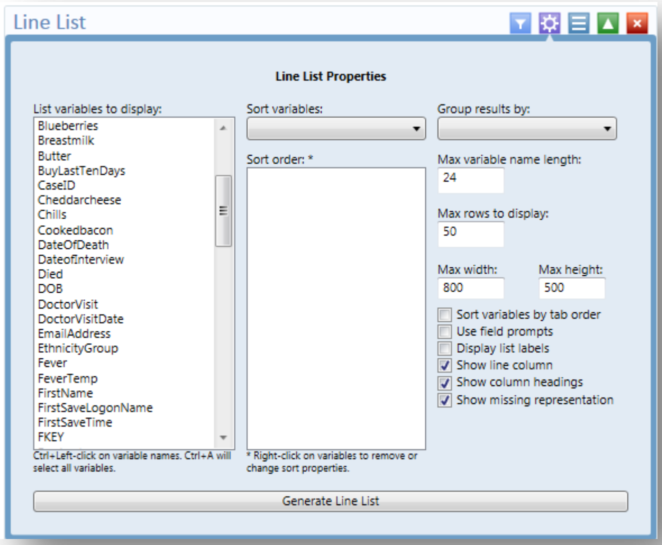
Figure 7.48: Line List Hidden View

Remove Sort Criteria

Remove sort criteria by following the steps below.

1. Right click on the **Line List.**
2. Select **Remove All Sort Criteria.**
3. The Line List will appear with the sort criteria removed.

Other Line List Properties



Line List

Line List Properties

List variables to display:
Blueberries
Breastmilk
Butter
BuyLastTenDays
CaseID
Cheddarcheese
Chills
Cookedbacon
DateOfDeath
DateofInterview
Died
DOB
DoctorVisit
DoctorVisitDate
EmailAddress
EthnicityGroup
Fever
FeverTemp
FirstName
FirstSaveLogonName
FirstSaveTime
FKEY

Ctrl+Left-click on variable names. Ctrl+A will select all variables.

Sort variables:

Sort order: *

* Right-click on variables to remove or change sort properties.

Group results by:

Max variable name length: 24

Max rows to display: 50

Max width: 800

Max height: 500

☐ Sort variables by tab order
☐ Use field prompts
☐ Display list labels
☑ Show line column
☑ Show column headings
☑ Show missing representation

Figure 7.49: Other Line List properties

- The **Max Variable Name Length** text box allows truncating long column names in the list output. It is set to 24 by default.

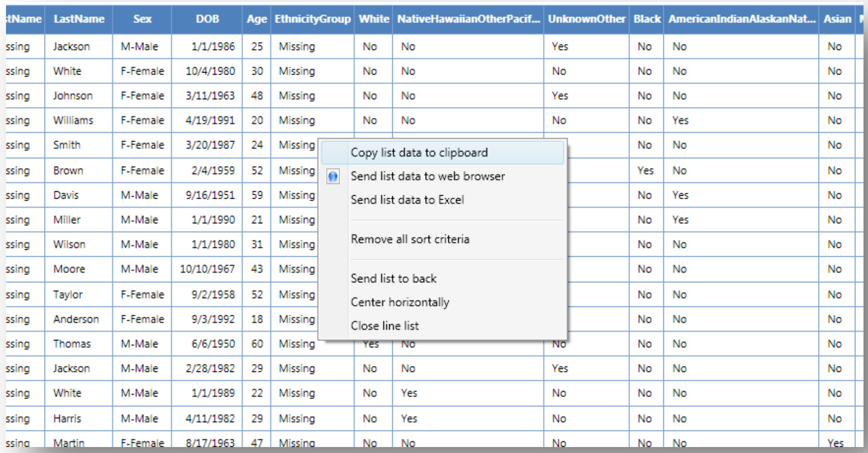
- The **Max Rows to Display** text box allows changing the maximum number of rows that the line list gadget will display. It is set to 50 by default. A maximum of 2000 rows may be displayed at any given time.
- The **Sort Variables by Tab Order** checkbox forces the columns in the output to be sorted by their tab order.* It is not checked by default.
- The **Use Field Prompts** checkbox will use the field's prompt as the column heading, rather than the field's name.* It is not checked by default.
- The **Display List Labels** checkbox will use the label value for option fields and comment legal fields rather than the underlying value stored in the database.*
  * Note: These features are available only when the Data Source is an Epi Info 7 project. If the Data Source is not an Epi Info 7 project, such as an Excel spreadsheet, these options have no effect.
- The **Show Line Column** option adds the Line column to the list and displays the line number when the checkbox is selected.
- The **Show Column Headings** option displays the line list column headings when the checkbox is selected.
- The **Show Missing Representation** option displays the word or symbol that represents a missing value for any field that does not have a value. By default, the word "Missing" is used, but other options include "Unknown", "(.)" or another custom setting as specified in the Tools > Options > Analysis tab. If the checkbox is not selected, a field without a value will be left blank.

Exporting a Line List

Once the Line List is created, the following export options are available: Copy List Data to Clipboard, Send List Data to Web Browser, and Send List Data to Excel.

Follow the steps below to Copy List Data to Clipboard.

1. Right click on the **Line List**.
2. Select **Copy List Data to Clipboard.**



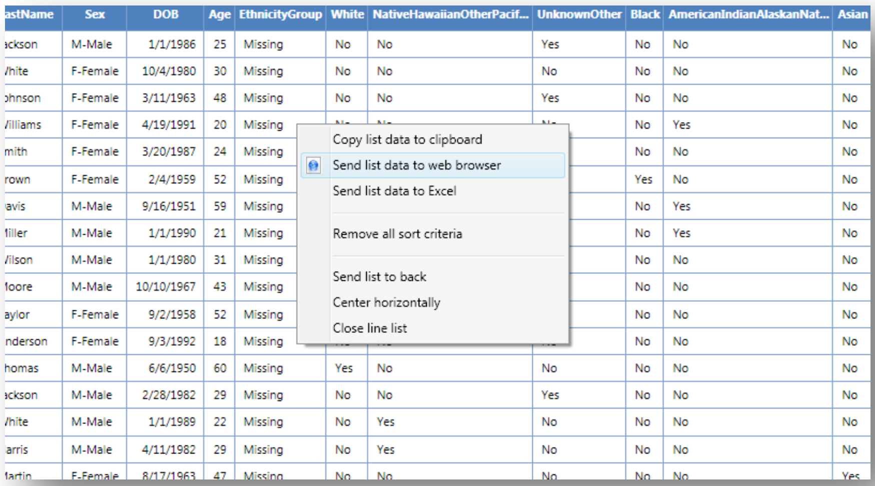
tName	LastName	Sex	DOB	Age	EthnicityGroup	White	NativeHawaiianOtherPacif...	UnknownOther	Black	AmericanIndianAlaskanNat...	Asian
ssing	Jackson	M-Male	1/1/1986	25	Missing	No	No	Yes	No	No	No
ssing	White	F-Female	10/4/1980	30	Missing	No	No	No	No	No	No
ssing	Johnson	F-Female	3/11/1963	48	Missing	No	No	Yes	No	No	No
ssing	Williams	F-Female	4/19/1991	20	Missing	No	No	No	No	Yes	No
ssing	Smith	F-Female	3/20/1987	24	Missing				No	No	No
ssing	Brown	F-Female	2/4/1959	52	Missing				Yes	No	No
ssing	Davis	M-Male	9/16/1951	59	Missing				No	Yes	No
ssing	Miller	M-Male	1/1/1990	21	Missing				No	Yes	No
ssing	Wilson	M-Male	1/1/1980	31	Missing				No	No	No
ssing	Moore	M-Male	10/10/1967	43	Missing				No	No	No
ssing	Taylor	F-Female	9/2/1958	52	Missing				No	No	No
ssing	Anderson	F-Female	9/3/1992	18	Missing				No	No	No
ssing	Thomas	M-Male	6/6/1950	60	Missing	Yes	No	No	No	No	No
ssing	Jackson	M-Male	2/28/1982	29	Missing	No	No	Yes	No	No	No
ssing	White	M-Male	1/1/1989	22	Missing	No	Yes	No	No	No	No
ssing	Harris	M-Male	4/11/1982	29	Missing	No	Yes	No	No	No	No
ssing	Martin	F-Female	8/17/1963	47	Missing	No	No	No	No	No	Yes

Figure 7.50: Copy List to Clipboard

3. The data is now on the clipboard and may be exported.

Follow the steps below to **Send List Data to Web Browser.**

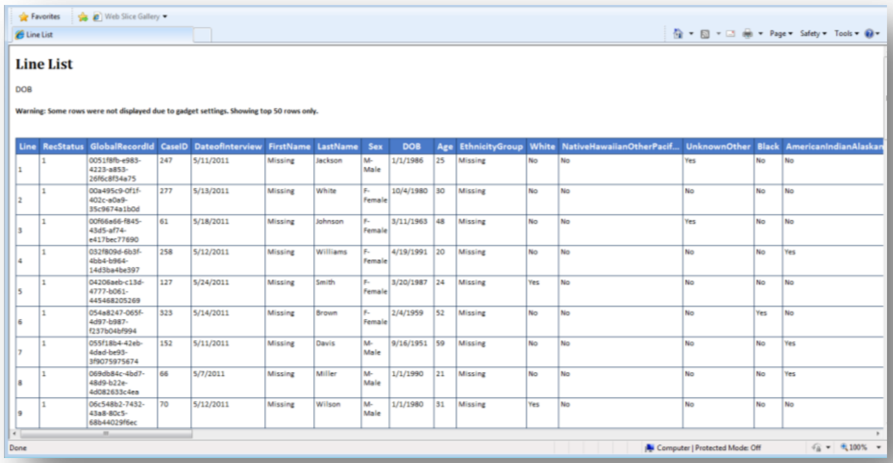
1. Right click on the **Line List**.
2. Select **Send List Data to Web Browser.**



astName	Sex	DOB	Age	EthnicityGroup	White	NativeHawaiianOtherPacif...	UnknownOther	Black	AmericanIndianAlaskanNat...	Asian
ackson	M-Male	1/1/1986	25	Missing	No	No	Yes	No	No	No
/hite	F-Female	10/4/1980	30	Missing	No	No	No	No	No	No
ohnson	F-Female	3/11/1963	48	Missing	No	No	Yes	No	No	No
/illiams	F-Female	4/19/1991	20	Missing				No	Yes	No
mith	F-Female	3/20/1987	24	Missing				No	No	No
rown	F-Female	2/4/1959	52	Missing				Yes	No	No
avis	M-Male	9/16/1951	59	Missing				No	Yes	No
liller	M-Male	1/1/1990	21	Missing				No	Yes	No
/ilson	M-Male	1/1/1980	31	Missing				No	No	No
loore	M-Male	10/10/1967	43	Missing				No	No	No
aylor	F-Female	9/2/1958	52	Missing				No	No	No
nderson	F-Female	9/3/1992	18	Missing				No	No	No
homas	M-Male	6/6/1950	60	Missing	Yes	No	No	No	No	No
ackson	M-Male	2/28/1982	29	Missing	No	No	Yes	No	No	No
/hite	M-Male	1/1/1989	22	Missing	No	Yes	No	No	No	No
arris	M-Male	4/11/1982	29	Missing	No	Yes	No	No	No	No
lartin	F-Female	8/17/1963	47	Missing	No	No	No	No	No	Yes

Figure 7.51: Send List Data to Web Browser

3. The data now appears in your web browser.



Line List

DOB

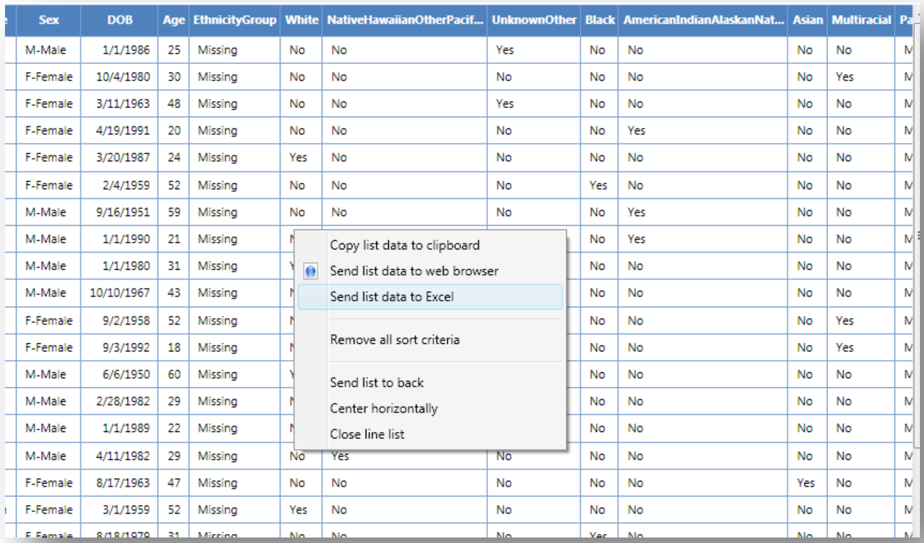
Warning: Some rows were not displayed due to gadget settings. Showing top 50 rows only.

Line	RecStatus	GlobalRecordId	CaseID	DateofInterview	FirstName	LastName	Sex	DOB	Age	EthnicityGroup	White	NativeHawaiianOtherPacif...	UnknownOther	Black	AmericanIndianAlaskan
1	1	0051f8fb-e983-4223-a853-26f6c8f34a75	247	5/11/2011	Missing	Jackson	M-Male	1/1/1986	25	Missing	No	No	Yes	No	No
2	1	00a495c9-0f1f-402c-a0a9-35c9674a1b0d	277	5/13/2011	Missing	White	F-Female	10/4/1980	30	Missing	No	No	No	No	No
3	1	00f66a66-f845-43d5-af74-e417bec77690	61	5/18/2011	Missing	Johnson	F-Female	3/11/1963	48	Missing	No	No	Yes	No	No
4	1	032f809d-6b3f-4bb4-b964-14d3ba4be397	258	5/12/2011	Missing	Williams	F-Female	4/19/1991	20	Missing	No	No	No	No	Yes
5	1	04206aeb-c13d-4777-b061-445468205269	127	5/24/2011	Missing	Smith	F-Female	3/20/1987	24	Missing	Yes	No	No	No	No
6	1	054a8247-065f-4d97-b987-f237b04bf994	323	5/14/2011	Missing	Brown	F-Female	2/4/1959	52	Missing	No	No	No	Yes	No
7	1	055f18b4-42eb-4dad-be93-3f9075975674	152	5/11/2011	Missing	Davis	M-Male	9/16/1951	59	Missing	No	No	No	No	Yes
8	1	069db84c-4bd7-48d9-b22e-4d082633c4ea	66	5/7/2011	Missing	Miller	M-Male	1/1/1990	21	Missing	No	No	No	No	Yes
9	1	06c548b2-7432-43a8-80c5-68b44029f6ec	70	5/12/2011	Missing	Wilson	M-Male	1/1/1980	31	Missing	Yes	No	No	No	No

Figure 7.52: List Data in Web Browser

Follow the steps below to **Send List Data to Excel.**

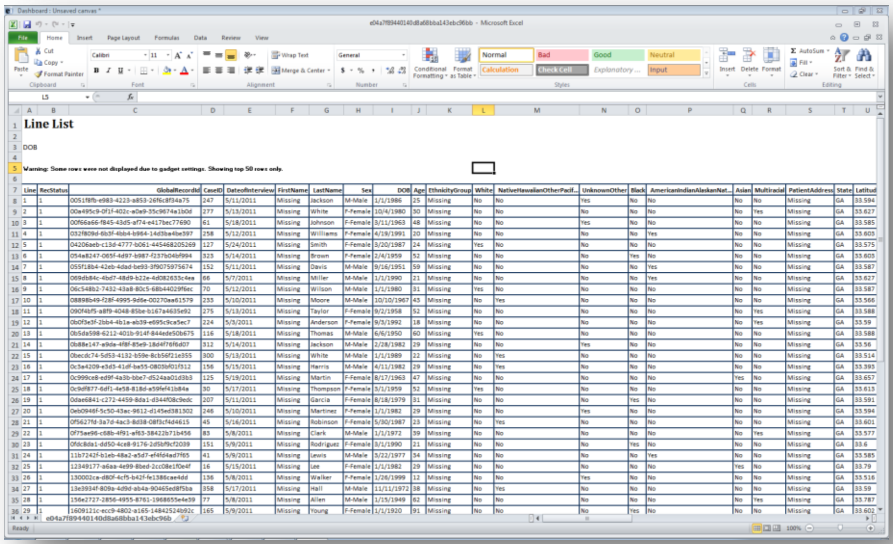
1. Right click on the **Line List**.
2. Select **Send List Data to Excel.**



Sex	DOB	Age	EthnicityGroup	White	NativeHawaiianOtherPacif...	UnknownOther	Black	AmericanIndianAlaskanNat...	Asian	Multiracial	Pa
M-Male	1/1/1986	25	Missing	No	No	Yes	No	No	No	No	M
F-Female	10/4/1980	30	Missing	No	No	No	No	No	No	Yes	M
F-Female	3/11/1963	48	Missing	No	No	Yes	No	No	No	No	M
F-Female	4/19/1991	20	Missing	No	No	No	No	Yes	No	No	M
F-Female	3/20/1987	24	Missing	Yes	No	No	No	No	No	No	M
F-Female	2/4/1959	52	Missing	No	No	No	Yes	No	No	No	M
M-Male	9/16/1951	59	Missing	No	No	No	No	Yes	No	No	M
M-Male	1/1/1990	21	Missing	N			No	Yes	No	No	M
M-Male	1/1/1980	31	Missing	Y			No	No	No	No	M
M-Male	10/10/1967	43	Missing	N			No	No	No	No	M
F-Female	9/2/1958	52	Missing	N			No	No	No	Yes	M
F-Female	9/3/1992	18	Missing	N			No	No	No	Yes	M
M-Male	6/6/1950	60	Missing	Y			No	No	No	No	M
M-Male	2/28/1982	29	Missing	N			No	No	No	No	M
M-Male	1/1/1989	22	Missing	N			No	No	No	No	M
M-Male	4/11/1982	29	Missing	No	Yes	No	No	No	No	No	M
F-Female	8/17/1963	47	Missing	No	No	No	No	No	Yes	No	M
F-Female	3/1/1959	52	Missing	Yes	No	No	No	No	No	No	M
F-Female	8/18/1979	31	Missing	No	No	No	Yes	No	No	No	

Figure 7.53: Send List Data to Excel

3. The list data appears in a Microsoft Excel spreadsheet.



Line List

DOB

Warning: Some rows were not displayed due to gadget settings. Showing top 50 rows only.

Line	RecStatus	GlobalRecordId	CaseID	DateofInterview	FirstName	LastName	Sex	DOB	Age	EthnicityGroup	White	NativeHawaiianOtherPacif...	UnknownOther	Black	AmericanIndianAlaskanNat...	Asian	Multiracial	PatientAddress	State	Latitud
1	1	0051f8fb-e983-4223-a853-26f6c8f34a75	247	5/11/2011	Missing	Jackson	M-Male	1/1/1986	25	Missing	No	No	Yes	No	No	No	No	Missing	GA	33.594
2	1	00a495c9-0f1f-402c-a0a9-35c9674a1b0d	277	5/13/2011	Missing	White	F-Female	10/4/1980	30	Missing	No	No	No	No	No	No	Yes	Missing	GA	33.627
3	1	00f66a66-f845-43d5-af74-e417bec77690	61	5/18/2011	Missing	Johnson	F-Female	3/11/1963	48	Missing	No	No	Yes	No	No	No	No	Missing	GA	33.585
4	1	032f809d-6b3f-4bb4-b964-14d3ba4be397	258	5/12/2011	Missing	Williams	F-Female	4/19/1991	20	Missing	No	No	No	No	Yes	No	No	Missing	GA	33.603
5	1	04206aeb-c13d-4777-b061-445468205269	127	5/24/2011	Missing	Smith	F-Female	3/20/1987	24	Missing	Yes	No	No	No	No	No	No	Missing	GA	33.575
6	1	054a8247-065f-4d97-b987-f237b04bf994	323	5/14/2011	Missing	Brown	F-Female	2/4/1959	52	Missing	No	No	No	Yes	No	No	No	Missing	GA	33.603
7	1	055f18b4-42eb-4dad-be93-3f9075975674	152	5/11/2011	Missing	Davis	M-Male	9/16/1951	59	Missing	No	No	No	No	Yes	No	No	Missing	GA	33.587
8	1	069db84c-4bd7-48d9-b22e-4d082633c4ea	66	5/7/2011	Missing	Miller	M-Male	1/1/1990	21	Missing	No	No	No	No	Yes	No	No	Missing	GA	33.627
9	1	06c548b2-7432-43a8-80c5-68b44029f6ec	70	5/12/2011	Missing	Wilson	M-Male	1/1/1980	31	Missing	Yes	No	No	No	No	No	No	Missing	GA	33.587
10	1	08898b49-f28f-4995-9d6e-00270aa61579	233	5/10/2011	Missing	Moore	M-Male	10/10/1967	43	Missing	No	Yes	No	No	No	No	No	Missing	GA	33.566
11	1	090f4bf5-a8f9-4048-85be-b167a4635e92	275	5/13/2011	Missing	Taylor	F-Female	9/2/1958	52	Missing	No	No	No	No	No	No	Yes	Missing	GA	33.588
12	1	0b0f3e3f-2bb4-4b1a-ab39-e695c9ca5ec7	224	5/3/2011	Missing	Anderson	F-Female	9/3/1992	18	Missing	No	No	No	No	No	No	Yes	Missing	GA	33.59
13	1	0b5da598-6212-401b-914f-844ede50b675	116	5/18/2011	Missing	Thomas	M-Male	6/6/1950	60	Missing	Yes	No	No	No	No	No	No	Missing	GA	33.588
14	1	0b88e147-a9da-4f8f-85e9-18d4f76f6d07	312	5/14/2011	Missing	Jackson	M-Male	2/28/1982	29	Missing	No	No	Yes	No	No	No	No	Missing	GA	33.56
15	1	0becdc74-5d53-4132-b59e-8cb56f21e355	300	5/13/2011	Missing	White	M-Male	1/1/1989	22	Missing	No	Yes	No	No	No	No	No	Missing	GA	33.514
16	1	0c3a4209-e3d3-41df-ba55-0803bf01f312	156	5/15/2011	Missing	Harris	M-Male	4/11/1982	29	Missing	No	Yes	No	No	No	No	No	Missing	GA	33.393
17	1	0c999ce8-ed9f-4a3b-bbe7-d524aa01d3b3	125	5/19/2011	Missing	Martin	F-Female	8/17/1963	47	Missing	No	No	No	No	No	Yes	No	Missing	GA	33.657
18	1	0c9df877-6df1-4e58-818d-a59fef41b84a	30	5/17/2011	Missing	Thompson	F-Female	3/1/1959	52	Missing	Yes	No	No	No	No	No	No	Missing	GA	33.613
19	1	0dae6841-c272-4459-8da1-d344f08c9edc	207	5/11/2011	Missing	Garcia	F-Female	8/18/1979	31	Missing	No	No	No	Yes	No	No	No	Missing	GA	33.591
20	1	0eb0946f-5c50-43ac-9612-d145ed381302	246	5/10/2011	Missing	Martinez	F-Female	1/1/1982	29	Missing	No	No	Yes	No	No	No	No	Missing	GA	33.594
21	1	0f5627fd-3a7d-4ac3-8d38-08f3cf4d4615	45	5/16/2011	Missing	Robinson	F-Female	5/30/1987	23	Missing	No	Yes	No	No	No	No	No	Missing	GA	33.601
22	1	0f75ae96-c68b-4f91-af63-38422b71b456	83	5/8/2011	Missing	Clark	M-Male	1/1/1972	39	Missing	No	No	No	No	No	No	Yes	Missing	GA	33.577
23	1	0fdc8da1-dd50-4ce8-9176-2d5bf9cf2039	151	5/9/2011	Missing	Rodriguez	F-Female	3/1/1990	21	Missing	No	No	No	Yes	No	No	No	Missing	GA	33.6
24	1	11b7242f-b1eb-48a2-a5d7-ef4fd4ad7f65	41	5/9/2011	Missing	Lewis	M-Male	3/22/1977	34	Missing	No	No	No	No	Yes	No	No	Missing	GA	33.585
25	1	12349177-a6aa-4e99-8bed-2cc08e1f0e4f	16	5/15/2011	Missing	Lee	F-Female	1/1/1982	29	Missing	No	No	No	No	No	Yes	No	Missing	GA	33.79
26	1	130002ca-d80f-4cf5-b42f-fe1386cae4dd	136	5/8/2011	Missing	Walker	F-Female	1/26/1999	12	Missing	No	No	Yes	No	No	No	No	Missing	GA	33.516
27	1	13e3934f-809a-4d9d-ab4a-90465ed8f5ba	358	5/17/2011	Missing	Hall	M-Male	11/11/1972	38	Missing	No	Yes	No	No	No	No	No	Missing	GA	33.59
28	1	156e2727-2856-4955-8761-1968655e4e39	77	5/8/2011	Missing	Allen	M-Male	1/15/1949	62	Missing	No	No	No	No	No	No	Yes	Missing	GA	33.787
29	1	1609121c-ecc9-4802-a165-14842524b92c	165	5/9/2011	Missing	Young	F-Female	1/1/1920	91	Missing	No	No	No	Yes	No	No	No	Missing	GA	33.602

e04a7f89440140d8a68bba143ebc96b

Figure 7.54: List Data in Excel

Note: You must have Microsoft Excel installed to use this feature.

Analysis Gadget > Frequency

The Frequency gadget counts each occurrence in a category for a specified variable and gives the absolute and relative frequencies for each category. This option then produces a frequency table that shows how many records have a value for each variable, the percentage of the total, a cumulative percentage and upper and lower confidence intervals.

1. Select the **EColi.PRJ** Data Source. Open the **FoodHistory** form from the **Data Source Explorer** menu.
2. Right click on the canvas and select **Add Analysis Gadget** > **Frequency**. The frequency gadget appears on the Visual Dashboard canvas.

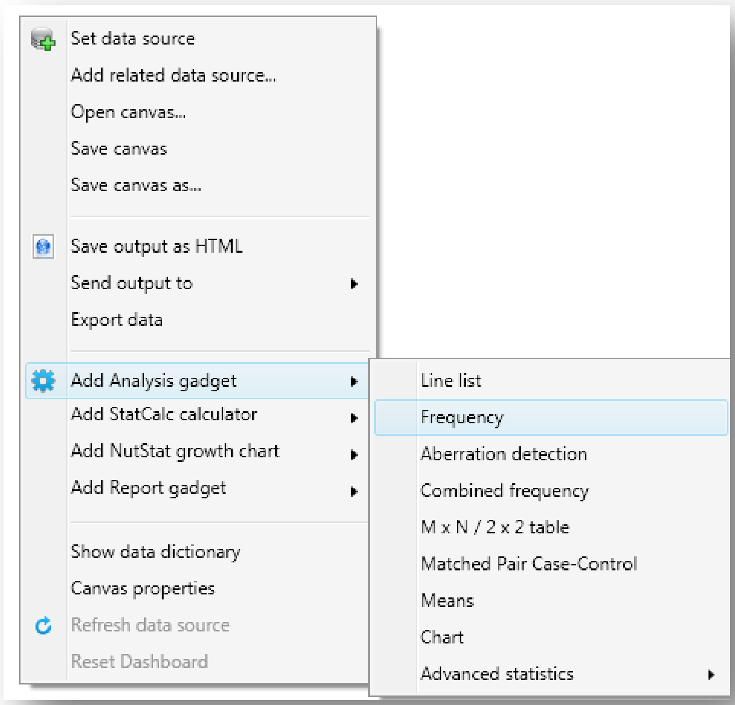


Figure 7.55: Frequency Menu Option

3. From the **Frequency Of** drop-down list, select a **variable** from the data table.

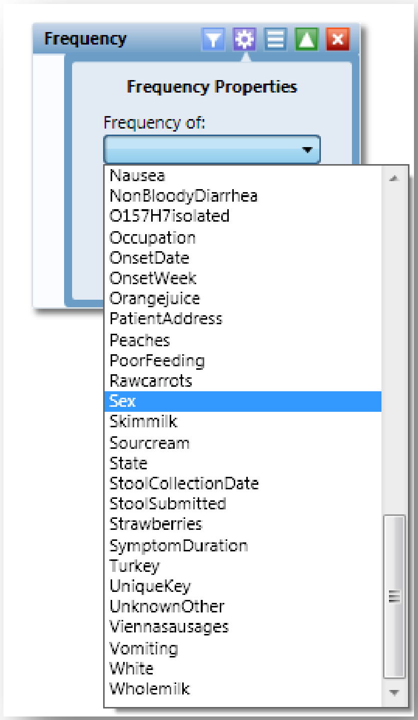
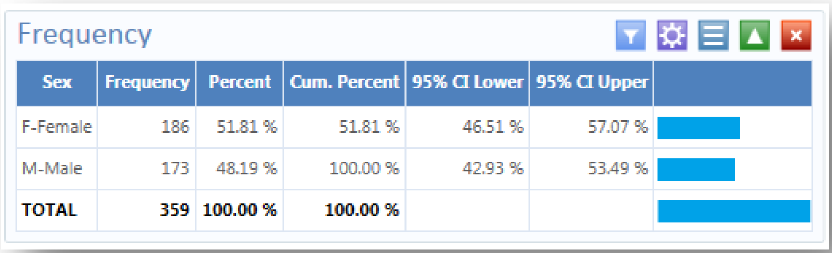


Figure 7.56: Selecting Frequency Of variable

4. Click **Run**. The results appear on the Visual Dashboard canvas.


Frequency

Sex	Frequency	Percent	Cum. Percent	95% CI Lower	95% CI Upper	
F-Female	186	51.81 %	51.81 %	46.51 %	57.07 %	
M-Male	173	48.19 %	100.00 %	42.93 %	53.49 %	
TOTAL	**359**	**100.00 %**	**100.00 %**			

Figure 7.57: Frequency Results

The **Frequency** column provides the count of individuals that are either female or male. The Percent column indicates the percentage of female or male. The **95% Confidence Limits** are a range of values that indicates the likely location of the true value of a measure, meaning (in this instance) that the percentage of females is likely to be between 46.51% and 57.07%.

Frequency Advanced Options

To use the **Frequency Advanced** options:

1. Select the **EColi.PRJ** Data Source. Open the **FoodHistory** form from the **Data Source Explorer** menu.
2. Right click on the canvas and select **Add Analysis Gadget** > **Frequency**. The frequency gadget appears on the Visual Dashboard canvas.
3. Click on the **Advanced options** arrow. The **Frequency Properties** dialog box appears.

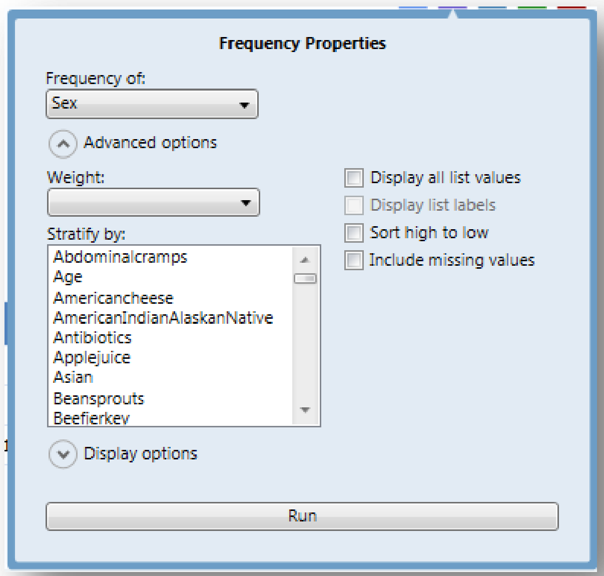


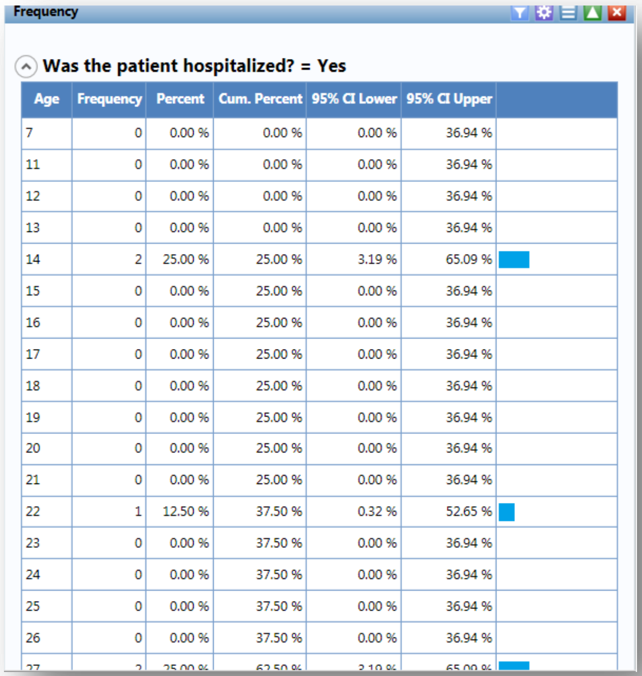
Figure 7.58: Frequency Properties

The following options are available in Frequency Properties:

- **Weight-** The field to use as the weight variable, for example if aggregate data is being analyzed.
- **Stratify By-** The field to use for stratifying the output.
- **Display all list values-** Only applicable when the frequency variable is a drop-down list field in an Epi Info™ 7 project. When selected, the output will show all of the drop-down list values even if they have a count of zero.
- **Display list labels** – Only applicable when the frequency variable is a Comment Legal field or Option Field in an Epi Info™ 7 project. When selected, the output will show the value and the label for Comment Legal fields and the label for Option fields.
- **Sort high to low** – displays the results sorted from high to low based on the selected Frequency of variable
- **Include Missing Values** – This option, which is unchecked by default, displays 'Missing' for any field with missing values.

4. From the **Frequency of** drop-down list, select **Age** from the data table.
5. From the **Weight** drop-down list, select **Died** from the data table.
6. Select **Hospitalized** from the **Stratify by** list.

7. Click **OK**. Results appear on the canvas.

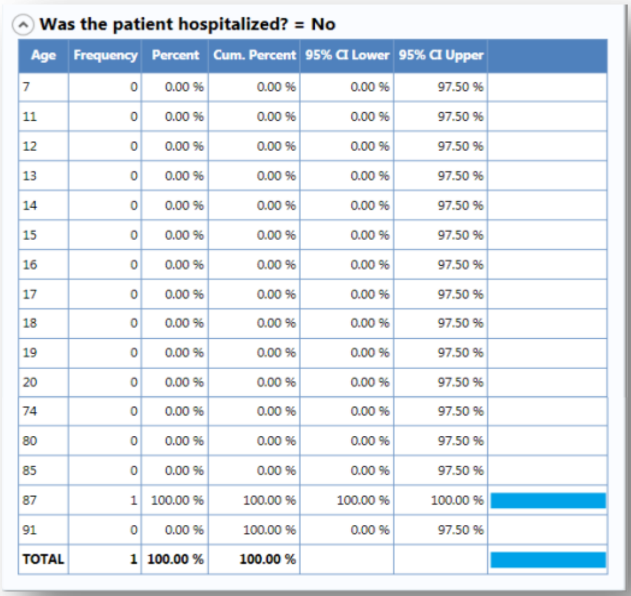


Frequency

Was the patient hospitalized? = Yes

Age	Frequency	Percent	Cum. Percent	95% CI Lower	95% CI Upper	
7	0	0.00 %	0.00 %	0.00 %	36.94 %	
11	0	0.00 %	0.00 %	0.00 %	36.94 %	
12	0	0.00 %	0.00 %	0.00 %	36.94 %	
13	0	0.00 %	0.00 %	0.00 %	36.94 %	
14	2	25.00 %	25.00 %	3.19 %	65.09 %	▬
15	0	0.00 %	25.00 %	0.00 %	36.94 %	
16	0	0.00 %	25.00 %	0.00 %	36.94 %	
17	0	0.00 %	25.00 %	0.00 %	36.94 %	
18	0	0.00 %	25.00 %	0.00 %	36.94 %	
19	0	0.00 %	25.00 %	0.00 %	36.94 %	
20	0	0.00 %	25.00 %	0.00 %	36.94 %	
21	0	0.00 %	25.00 %	0.00 %	36.94 %	
22	1	12.50 %	37.50 %	0.32 %	52.65 %	▬
23	0	0.00 %	37.50 %	0.00 %	36.94 %	
24	0	0.00 %	37.50 %	0.00 %	36.94 %	
25	0	0.00 %	37.50 %	0.00 %	36.94 %	
26	0	0.00 %	37.50 %	0.00 %	36.94 %	
27	2	25.00 %	62.50 %	3.19 %	65.09 %	▬

Figure 7.59: Frequency Line List Stratified View #1



Was the patient hospitalized? = No

Age	Frequency	Percent	Cum. Percent	95% CI Lower	95% CI Upper	
7	0	0.00 %	0.00 %	0.00 %	97.50 %	
11	0	0.00 %	0.00 %	0.00 %	97.50 %	
12	0	0.00 %	0.00 %	0.00 %	97.50 %	
13	0	0.00 %	0.00 %	0.00 %	97.50 %	
14	0	0.00 %	0.00 %	0.00 %	97.50 %	
15	0	0.00 %	0.00 %	0.00 %	97.50 %	
16	0	0.00 %	0.00 %	0.00 %	97.50 %	
17	0	0.00 %	0.00 %	0.00 %	97.50 %	
18	0	0.00 %	0.00 %	0.00 %	97.50 %	
19	0	0.00 %	0.00 %	0.00 %	97.50 %	
20	0	0.00 %	0.00 %	0.00 %	97.50 %	
74	0	0.00 %	0.00 %	0.00 %	97.50 %	
80	0	0.00 %	0.00 %	0.00 %	97.50 %	
85	0	0.00 %	0.00 %	0.00 %	97.50 %	
87	1	100.00 %	100.00 %	100.00 %	100.00 %	▬
91	0	0.00 %	100.00 %	0.00 %	97.50 %	
TOTAL	**1**	**100.00 %**	**100.00 %**			▬

Figure 7.60: Frequency Line List Stratified View #2

The results display the frequency of deceased patients by age and are stratified by whether the patient was hospitalized or not. The age, frequency, percent, cumulative percent and upper and lower 95% confidence intervals are displayed. The last column is the percent bar.

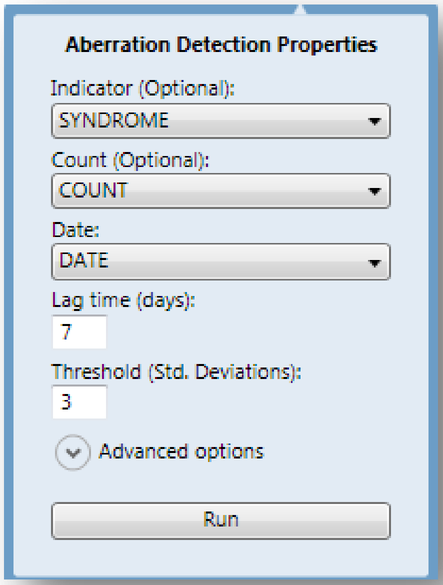
You can also click on the Display options arrow and expand the Frequency Properties. The following options are available in Frequency Properties:

- **Use field prompt** – specifies whether to use field prompts and is checked by default
- **Draw borders** – Setting for drawing the frequency's border.
- **Draw header row** – Setting for displaying the frequency's headers.
- **Draw total row** – Setting for displaying the frequency's totals.
- **Decimals to Display** – specifies the number of decimals that will be displayed in the results. The possible values range from zero to four. This is set to two by default.
- **Maximum rows to display** – Specifies number of rows to display.
- **Max width of percent bar** – Width for percent bar column using a % value.

Analysis Gadget > Aberration Detection

The Aberration Detection Analysis gadget can be used with syndromic or national surveillance data. It is designed to monitor changes in the distribution or frequency of health events when compared to historical data. This gadget uses an algorithm implemented by the Early Aberration Reporting System, which is a tool developed by CDC's Division of Bioterrorism Preparedness and Response to assist local-level health offices with early detection of bioterrorism events or possible outbreaks.

1. Select the **Ears.xls** Data Source from the Sample project folder. Click on **Sheet 1$**.
2. Right click on the canvas and select **Add Analysis Gadget** > **Aberration Detection**. The **Aberration Detection Properties** dialog box appears.



Aberration Detection Properties

Indicator (Optional): SYNDROME

Count (Optional): COUNT

Date: DATE

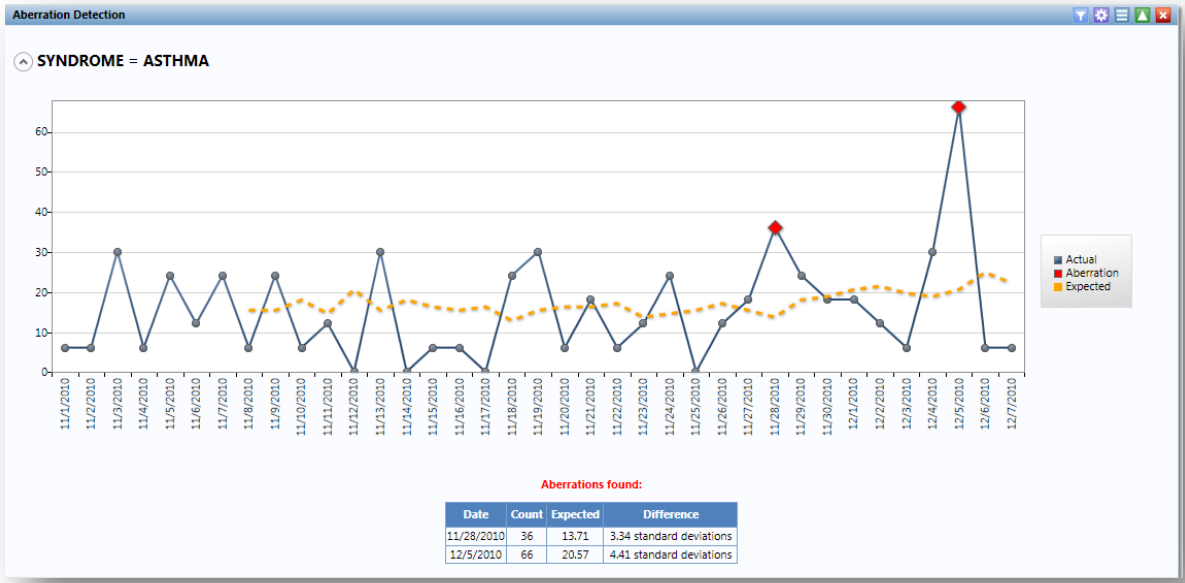
Lag time (days): 7

Threshold (Std. Deviations): 3

Figure 7.61: Aberration Detection Properties

3. Select **Syndrom**e from the **Indicator** drop-down list.

4. Select **Count** from the **Count (Optional)** drop-down list.
5. Select **Date** from the **Date** drop-down list.
6. Enter **7** for **Lag Time (days)**.
7. Enter **3** for **Threshold (Std. Deviations)**.
8. Click the **Run** button. The Aberration Detection line graph displays on the canvas.



Aberrations found:

Date	Count	Expected	Difference
11/28/2010	36	13.71	3.34 standard deviations
12/5/2010	66	20.57	4.41 standard deviations

Figure 7.62: Aberration Detection Line Graph

The result displays the aberrations found, date, count, expected count, and the standard deviations. The current number of patient visits with Asthma is higher (Count: 36, 66) than the defined threshold (Expected; 13.71, 20.57), hence the aberrations detected on 11/28/2010 and 12/05/2010. These aberrations may need to be evaluated by an epidemiologist (or other public health professional) to determine whether it signifies an event of public health importance warranting further investigation.

Analysis Gadget > Combined Frequency

The Combined Frequency option produces a table that illustrates the frequency and percentage of grouped variables.

1. Select the **EColi.PRJ** Data Source. Open the **FoodHistory** form from the **Data Source Explorer** menu.
2. Right click on the canvas and select **Add Analysis Gadget** > **Combined Frequency**. The combined frequency gadget appears on the Visual Dashboard canvas.

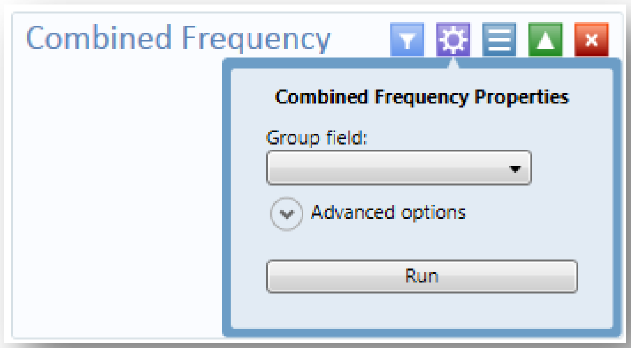
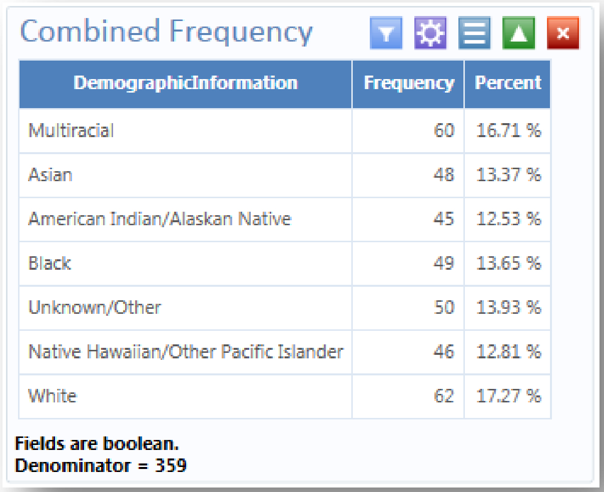


Figure 7.63: Combined Frequency Properties

3. From the **Group field** drop-down list, select a **DemographicInformation**.
4. Click **Run.** The results appear on the Visual Dashboard canvas.



Combined Frequency

DemographicInformation	Frequency	Percent
Multiracial	60	16.71 %
Asian	48	13.37 %
American Indian/Alaskan Native	45	12.53 %
Black	49	13.65 %
Unknown/Other	50	13.93 %
Native Hawaiian/Other Pacific Islander	46	12.81 %
White	62	17.27 %

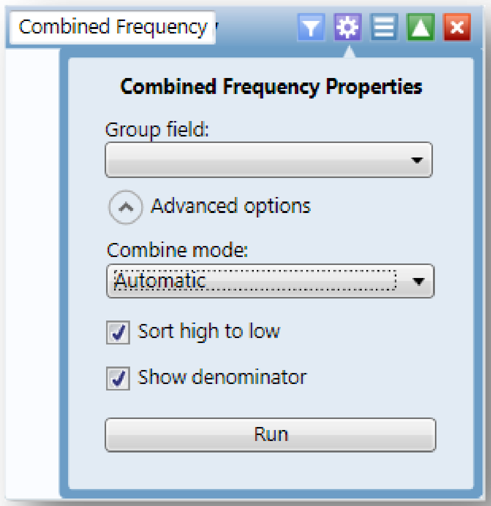
Fields are boolean.
Denominator = 359

Figure 7.64: Combined Frequency Results

Combined Frequency Advanced Options

To use the Combined Frequency Advanced options:

1. Select the **EColi.PRJ** Data Source. Open the **FoodHistory** form from the **Data Source Explorer** menu.
2. Right click on the canvas and select **Add Analysis Gadget** > **Combined Frequency**. The combined frequency gadget appears on the Visual Dashboard canvas.
3. Click on the arrow to display **Advanced Options.**



Combine mode: Automatic

☑ Sort high to low

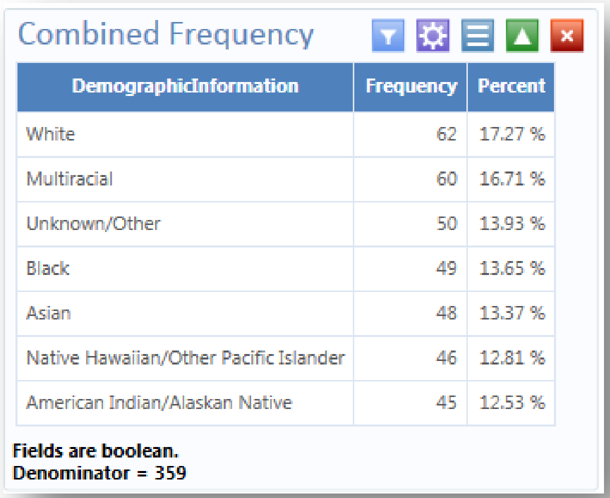
☑ Show denominator

Figure 7.65: Combined Frequency Properties Combine Mode

4. From the **Group Field** drop-down list, select **DemographicInformation**.
5. Select **Automatic** from the **Combine Mode** drop-down list.

   - Automatic- will automatically determine the appropriate combination mode of either Boolean or Categorical.
   - Boolean- use this mode for Yes/No and Checkbox fields or if you wish for the fields in the group to be treated as such.
   - Categorical- use this mode if your data is grouped according to similar characteristics in a way that shows the relative frequencies of each group or category.
   - Value to treat as true – This field appears when the Combine Mode is Boolean. The specified value will be treated as the true or positive value and all other values will be treated as false or negative values.

6. Mark your selection for **Sort High to Low.** This option will display the results sorted from highest group field frequency to the lowest. The **Sort High to Low** checkbox is checked by default.
7. Mark your selection for **Show Denominator**. This option will display the total number of records in the results. The **Show Denominator** checkbox is checked by default.
8. Click **Run**. The Combined Frequency results appear on the Visual Dashboard canvas.


Combined Frequency

DemographicInformation	Frequency	Percent
White	62	17.27 %
Multiracial	60	16.71 %
Unknown/Other	50	13.93 %
Black	49	13.65 %
Asian	48	13.37 %
Native Hawaiian/Other Pacific Islander	46	12.81 %
American Indian/Alaskan Native	45	12.53 %

Fields are boolean.
Denominator = 359

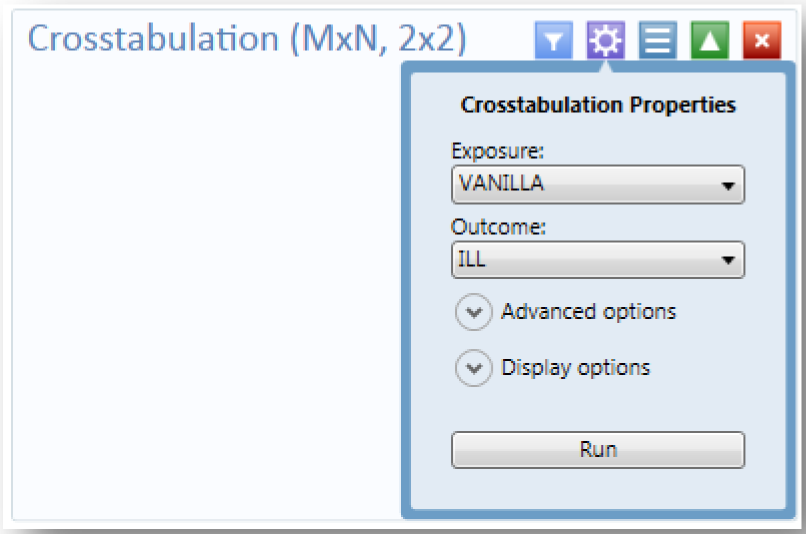
Figure 7.66: Combined Frequency Results

The result displays the frequency and percentage of each demographic category.

Analysis Gadget > M x N/ 2 x 2 Table
In epidemiology, 2 x 2 tables are frequently used to examine the relationship between two or more categorical values. In these tables, usually an exposure variable is considered the risk factor. The outcome variable is considered the disease of consequence (e.g., the person had the disease or outcome of interest or they did not). Values of the first variable will appear on the left margin of the table, and those of the second will be across the top of the table. Normally, cells contain counts of records matching the values in corresponding marginal labels. The M x N /2 x 2 Table analysis gadget in Visual Dashboard is demonstrated below:

1. Select the **Sample.PRJ** Data Source. Open the **Oswego** form from the **Data Source Explorer** menu. Click **OK**.
2. Right click on the canvas and select **Add Analysis Gadget** > **M x N /2 x 2 table**. The **Crosstabulation Properties** dialog box appears on the Visual Dashboard canvas.
3. From the **Exposure** drop-down list, select **Vanilla.**
4. From the **Outcome** drop-down list, select **ILL**.

Advanced Options allow you to; add a weight variable, select **Stratify by** variable and alter display settings. Advanced Options are not used in this example.

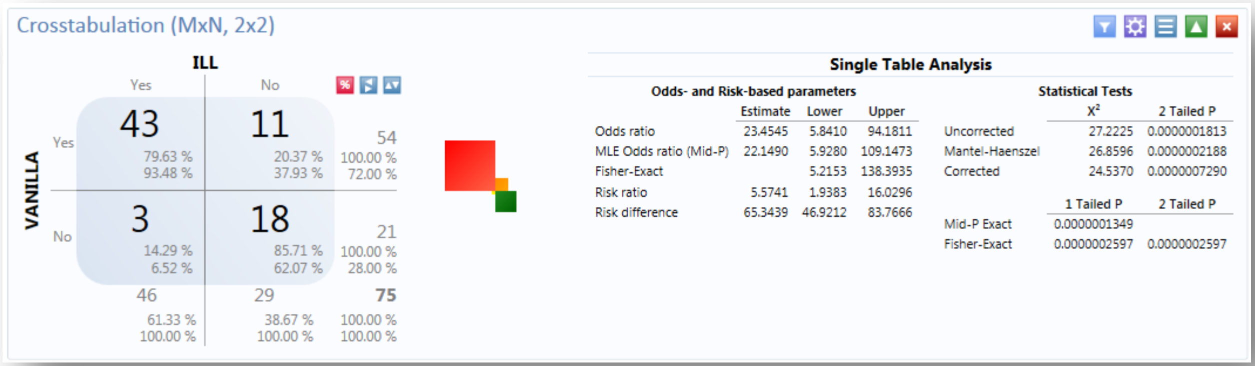


Exposure: VANILLA

Outcome: ILL

Figure 7.67: Crosstabulation Properties

5. Click **Run**. Results appear on the Visual Dashboard canvas.



VANILLA	ILL: Yes	ILL: No	
Yes	43 79.63 % 93.48 %	11 20.37 % 37.93 %	54 100.00 % 72.00 %
No	3 14.29 % 6.52 %	18 85.71 % 62.07 %	21 100.00 % 28.00 %
	46 61.33 % 100.00 %	29 38.67 % 100.00 %	**75** 100.00 % 100.00 %

Single Table Analysis

Odds- and Risk-based parameters

	Estimate	Lower	Upper
Odds ratio	23.4545	5.8410	94.1811
MLE Odds ratio (Mid-P)	22.1490	5.9280	109.1473
Fisher-Exact		5.2153	138.3935
Risk ratio	5.5741	1.9383	16.0296
Risk difference	65.3439	46.9212	83.7666

Statistical Tests

	X^2	2 Tailed P
Uncorrected	27.2225	0.0000001813
Mantel-Haenszel	26.8596	0.0000002188
Corrected	24.5370	0.0000007290

	1 Tailed P	2 Tailed P
Mid-P Exact	0.0000001349	
Fisher-Exact	0.0000002597	0.0000002597

Figure 7.68: Crosstabulation Results

The results display a 2 x 2 table and a single table analysis of all records cross-tabulated by exposure to vanilla where the outcome was ill. Out of the 75 records in this dataset, 43 patients were exposed to vanilla and became ill. Eleven patients were exposed to vanilla and did not become ill. Of the 21 patients that were not exposed to vanilla, only three became ill.

The Single Table Analysis chart on the right displays Odds Ratios, Risk Ratios, and Statistical Tests results for this example.

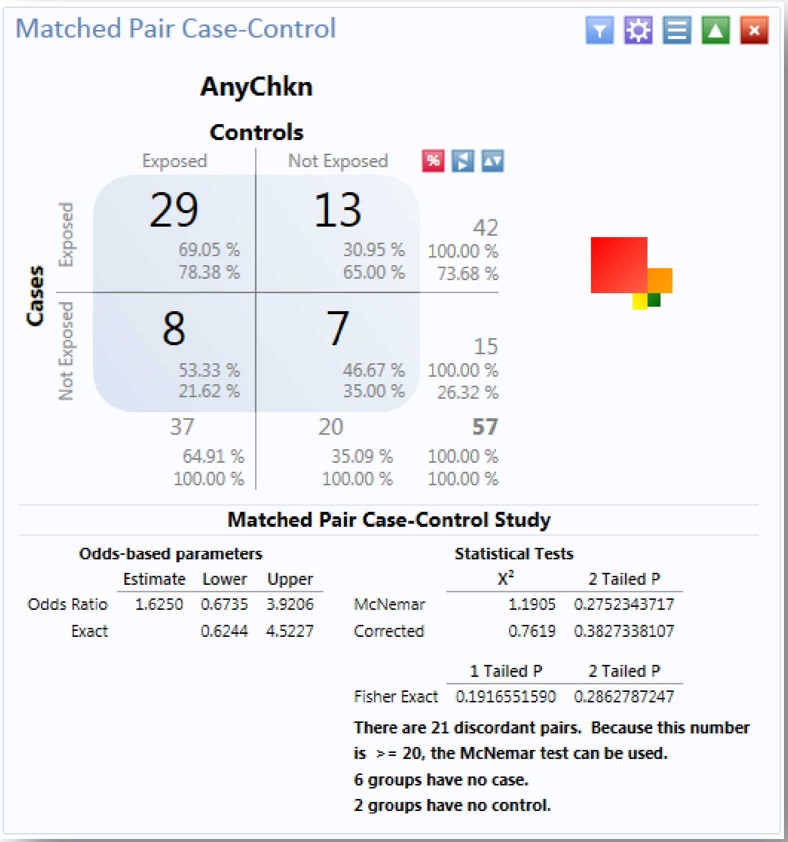
Analysis Gadget > Matched Pair-Case Control

The Matched Pair Case Control option in Visual Dashboard is for use with pair-matched case-control studies. Each case must have exactly one matched control. For this reason, it is generally incorrect to perform the analysis using 2x2 tables since it does not take matching into consideration. To use this option, you need to specify an Exposure Variable, the Outcome Variable (i.e., case vs. control), and a Match variable that links each case to its control. The following example demonstrates a matched pair-case control analysis.

The dataset below is from a matched case-control study of the consumption of chicken where the outcome resulted in the patient becoming ill. The primary exposure was the consumption of chicken (AnyChkn), the case control is **CaCO**, and the **Pair Group ID** is **Matched pairs**.

1. Select the **Case Control DatabaseExample.xlsx** Data Source.
2. Select **Section_1_3 & 8** form in the **Data Source Explorer** menu**.**
3. Click Ok.
4. Select **Add Analysis Gadget > Matched Pair-Case Control**.
5. The **Matched Pair Case Control Properties** dialog box appears.
6. From the **Exposure** drop-down list, select **AnyChkn_RECODED**.
7. From the **Case/Control** drop-down list, select **CaCo**.
8. From the **Pair Group ID** drop-down list, select **Matched pairs.**
9. Click on **Define Value Mappings**
10. Click on **1**
11. Click on the **>** button on the **Yes values:** section
12. Click on **0**
13. Click on the **>** button on the **No values:** section
14. Click **Ok**.
15. Click **Run**

The **Matched Pair Case-Control** results appear on the canvas.



Matched Pair Case-Control

AnyChkn

Controls

		Exposed	Not Exposed	
Cases	Exposed	29 69.05 % 78.38 %	13 30.95 % 65.00 %	42 100.00 % 73.68 %
	Not Exposed	8 53.33 % 21.62 %	7 46.67 % 35.00 %	15 100.00 % 26.32 %
		37 64.91 % 100.00 %	20 35.09 % 100.00 %	57 100.00 % 100.00 %

Matched Pair Case-Control Study

Odds-based parameters

	Estimate	Lower	Upper
Odds Ratio	1.6250	0.6735	3.9206
Exact		0.6244	4.5227

Statistical Tests

	X^2	2 Tailed P
McNemar	1.1905	0.2752343717
Corrected	0.7619	0.3827338107

	1 Tailed P	2 Tailed P
Fisher Exact	0.1916551590	0.2862787247

There are 21 discordant pairs. Because this number is >= 20, the McNemar test can be used.

6 groups have no case.

2 groups have no control.

Figure 7.69: Matched Pair Case-Control results

The results display that out of 57 matched pairs, 29 had both the case and the control exposed to AnyChkn, 13 had the case but not the control exposed, eight pairs had the control exposed but the case unexposed, and seven had neither the case nor the control exposed.

The Odds-Ratio is greater than 1, which suggests that eating chicken may be more likely to result in being ill. However, the confidence intervals contain 1 and the P values are less than 0.05, which indicate that there is no significant association between eating chicken and becoming ill.

Analysis Gadget > Means

The Means gadget calculates the average for a continuous numeric variable. A **Yes/No** field returns numeric values (Yes=1, No=0) which allows the **Means** gadget to calculate the proportion of respondents answering yes. For this situation, **Analysis Gadget > Frequency**, has two formats:

- If only one variable is selected, the program produces a table similar to one produced by the Frequency option with descriptive statistics.
- If two variables are selected, the first variable contains the data to be analyzed. The second indicates how groups will be distinguished. The output of this format is a table similar to one produced by the M x N/ 2 x 2 table with descriptive statistics.

1. Select the **Sample.PRJ** Data Source. Open the **Oswego** form from the **Data Source Explorer** menu.
2. Click **OK**.
3. Right click on the canvas and select **Add Analysis Gadget** > **Means**. The **Means Properties** dialog box appears on the Visual Dashboard canvas.

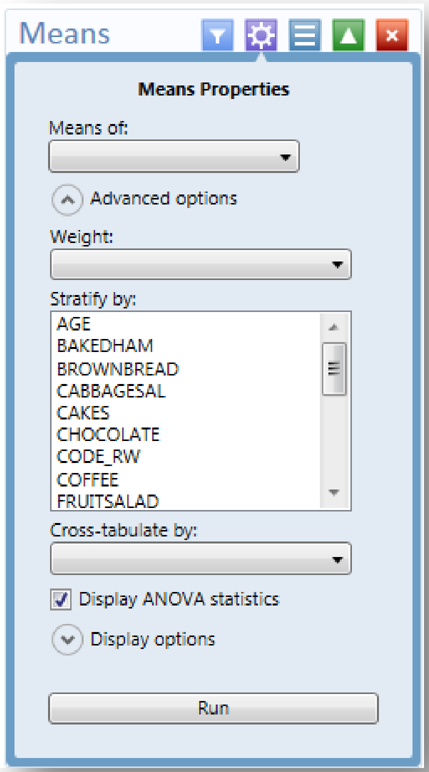
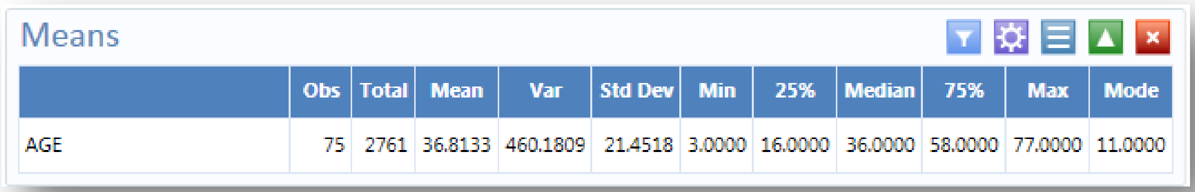


Figure 7.70: Means Properties

5. From the **Means Of** drop-down list, select **Age**.

6. Click the arrow to display **Advanced Options.**
   - The **Cross-tabulate by** value of drop-down list contains a variable to help determine if the means of a group are equal.
   - The **Stratify by** drop-down list contains a variable to act as a grouping variable.
   - The **Weight** drop-down list contains a variable for weighted analysis.
   - The **Output columns to display** option specifies the columns to include in this analysis.
   - The **Decimals to Display** specifies the number of decimals that will be displayed in the results. The possible values range from zero to four. This is set to two by default.
   - The **Display ANOVA statistics** checkbox specifies whether to display the analysis of variance statistics. This option is selected by default. If with large data sets, the ANOVA statistics are not required, performance may be improved by turning off (un-checking) this feature.

6. Click **Run**. The results are displayed on the canvas.


Means

	Obs	Total	Mean	Var	Std Dev	Min	25%	Median	75%	Max	Mode
AGE	75	2761	36.8133	460.1809	21.4518	3.0000	16.0000	36.0000	58.0000	77.0000	11.0000

Figure 7.71: Means Table

The Mean age of individuals in the dataset is 36.8.

Analysis Gadget > Chart

Visual Dashboard produces histograms, scatter plots, pie charts, bar and line graphs directly from data files. This option produces charts based on the types of data available in the project. The following are the graph types capable of being generated by the Chart option:

- **Epi Curve** charts use vertical bars to represent the count or weight for each value of the main variable. Each series creates an additional vertical bar at each point. The main variable must be either numeric or date/time. This graph differs from the bar graph because adjacent bars represent equal ranges of the main variable.
- **Scatter** charts display variables along an X-Y axis as a scatter plot. The X variable is the independent variable and Y variable is the dependent variable. Each series is represented by a different point style.
- **Stacked Column** is similar to a bar chart but represents the counts or weights of two variables.

- **Bar** charts use horizontal bars to represent the count or weight of each value of the main variable(s). Each series creates an additional horizontal bar at each point.
- **Column** charts use vertical bars to represent the count or weight for each value of the main variable(s). Each series results in an additional vertical bar at each point; the bars are distinguished by their style.
- **Line** charts connect X and Y data points with straight lines. Each series is represented by a different style of line and both variables must be numeric.
- **Pie** charts use a circle to represent each series and each value of the main variable has a slice of the circle proportional to the value associated with it.
- **Pareto** charts use vertical bars to represent the count or weight for each value of the main variable(s) and a dashed line to represent the accumulated percentage. Each series results in an additional vertical bar at each point; the bars are distinguished by their style.

The examples below demonstrate how to create a column chart, a pie chart, an Epi Curve chart, and a Pareto chart.

Generate a Column Chart

Follow the steps below to create a bar graph using data from the Sample.PRJ project.

1. Open the **Sample.PRJ** project and select the **Oswego** form.
2. Right click on the canvas and select **Add Analysis Gadget** > **Charts>Column chart**. The **Chart Properties** dialog box appears on the Visual Dashboard canvas.
3. From the **Main variable** drop-down list, select the variable **Age** to use for the independent variable.
4. Leave the **Weight**, **Stratify By**, **Chart size**, and **Show grid Lines** options at the default settings.
5. Click **Run**. The graph appears on the canvas displaying the number of patients by age.

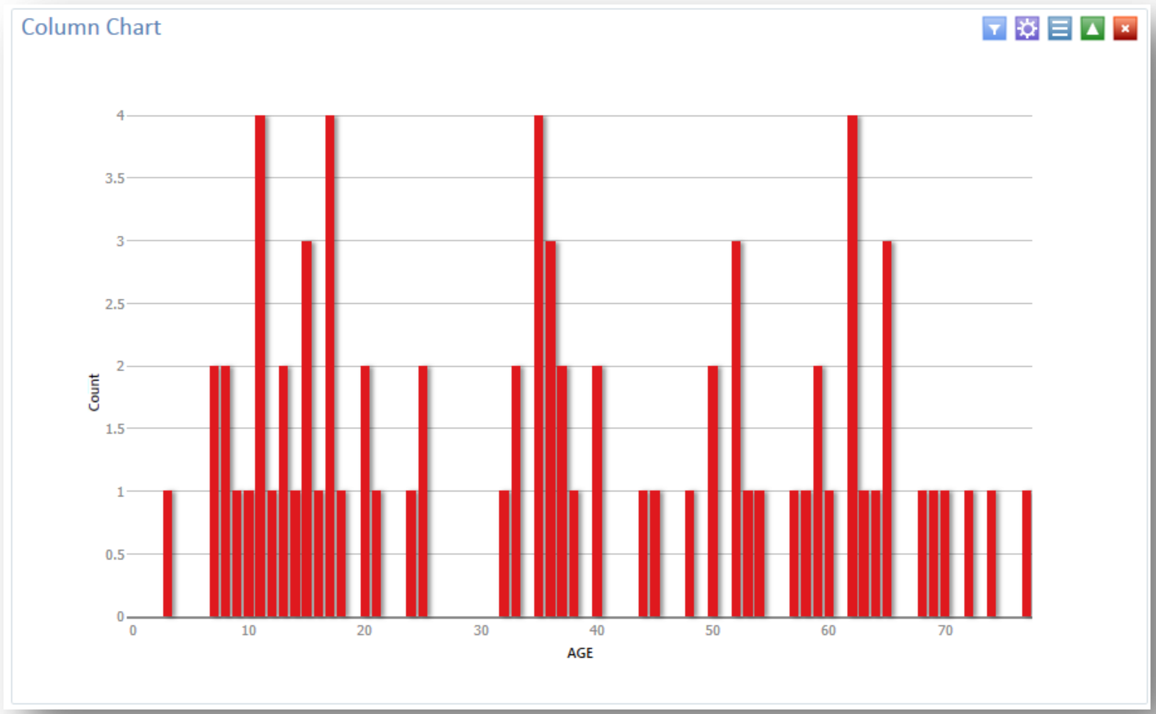


Figure 7.72: Column Chart

Generate a Pie Chart

Follow the steps below to create a pie chart showing age categories of church supper attendees.

1. Open the **Sample.PRJ** project and select the **Oswego** form. Recode the **Age** variable using the **Defined Variables** gadget.

   - On the left-hand side of the Visual Dashboard canvas, move the mouse cursor over the **Defined Variables** gadget. The gadget expands and becomes fully visible.
   - Click the **New Variable** button.
   - Select **With Recoded Value**. The **Add Recoded Variable** dialog box appears.
   - From the **Source** field drop-down list, select **Age**.
   - Click the **Fill Ranges** button. The **Fill Ranges** dialog box appears.
   - Enter **0** for **Start value**, **70** for **End value**, and **10** for **By**.
   - Click **OK**. The **Add Recoded Variable** window appears with the **From, To**, and **Representation** columns populated.
   - Click **OK**. The variables are recoded.

2. Right click on the canvas and select **Add Analysis Gadget > Charts>Pie chart**. The **Chart Properties** dialog box appears on the Visual Dashboard canvas.

3. From the **Main variable** drop-down list, select the variable **AGE_RECODED**.
4. Leave the **Weight**, **Chart Size**, **Show grid lines** options at the default settings.
5. Click **Run**. The graph appears on the canvas.

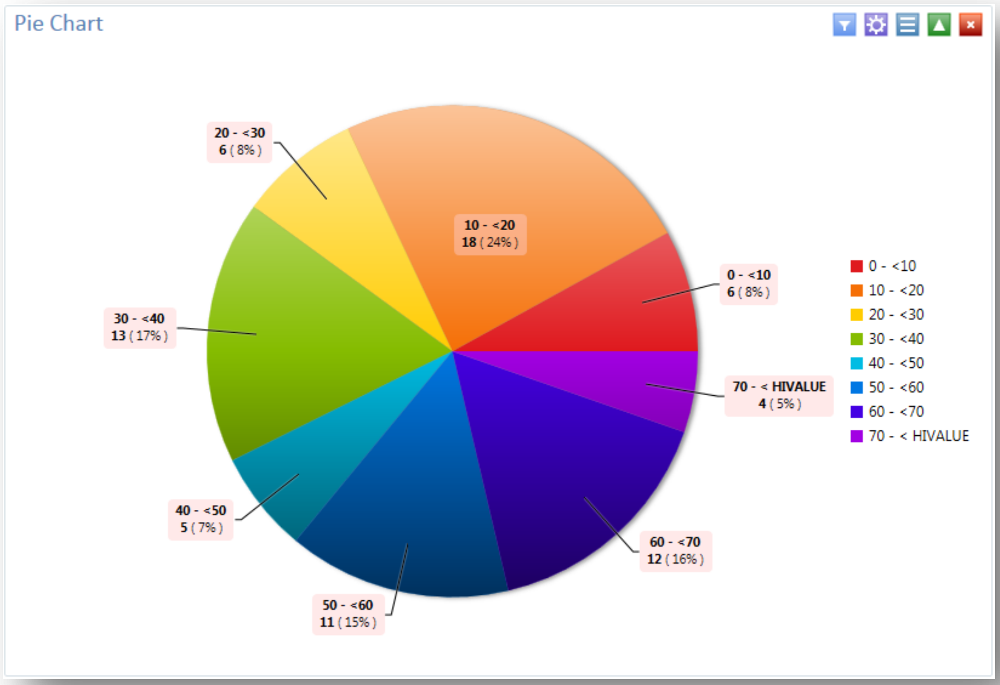


Figure 7.73: Pie Chart

The chart shows the age categories of church supper attendees by age group.

Generate an Epi Curve

Follow the steps below to create an Epi Curve showing the incubation time for an unknown pathogen.

1. Open the **Ecoli.PRJ** project and select the **FoodHistory** form.
2. Right click on the canvas and select **Add Analysis Gadget** > **Chart>Epi curve chart**. The **Chart Properties** dialog box appears on the Visual Dashboard canvas.
3. From the **Main variable** drop-down list, select **OnsetDate**.
4. Click on the Configuration panel icon.
5. Click the arrow to expand the **Display options**.
6. Update the **X-axis angle:** setting to -90. The date values should now be displayed on the x-axis. If your computer screen is too small, you might need to change the width setting of the chart to a higher number until the values are displayed.
7. Change the width setting to **1000**
8. Click on the **Show Legend** box
9. Click on the arrow to expand the **Advanced options**.
10. Click on the **Sex** variable in the **Stratify by:** section.

11. Click on the **Run** button. The Epi Curve chart appears on the canvas.

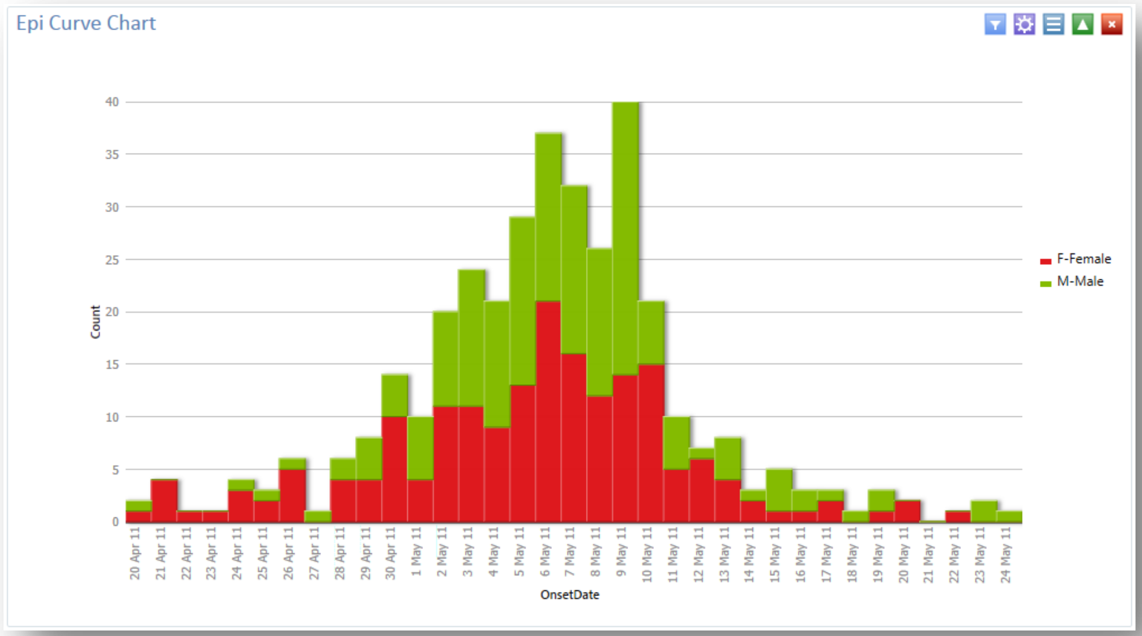


Figure 7.74: Epi Curve Chart

Generate a Pareto Chart

Follow the steps below to create an Epi Curve showing the incubation time for an unknown pathogen.

1. Open the **Ecoli.PRJ** project and select the **FoodHistory** form.
2. Recode the Age variable using the **Defined Variables** gadget.

   - On the left-hand side of the Visual Dashboard canvas, move the mouse cursor over the **Defined Variables** gadget. The gadget expands and becomes fully visible.
   - Click the **New Variable** button.
   - Select **With Recoded Value**. The **Add Recoded Variable** dialog box appears.
   - From the **Source** field drop-down list, select **Age**.
   - Click the **Fill Ranges** button. The **Fill Ranges** dialog box appears.
   - Enter **0** for **Start value**, **70** for **End value**, and **2** for **By**.
   - Click **OK**. The **Add Recoded Variable** window appears with the **From**, **To**, and **Representation** columns populated.
   - Click **OK**.

3. Right click on the canvas and select **Add Analysis Gadget** > **Charts>Pareto chart**. The **Chart Properties** dialog box appears on the Visual Dashboard canvas.
4. From the **Chart Type** drop-down list, select **Pareto**.

5. From the **Main variable** drop-down list, select the **Age_Recoded**.
6. Click on the **Run** button. The Pareto chart appears on the canvas.

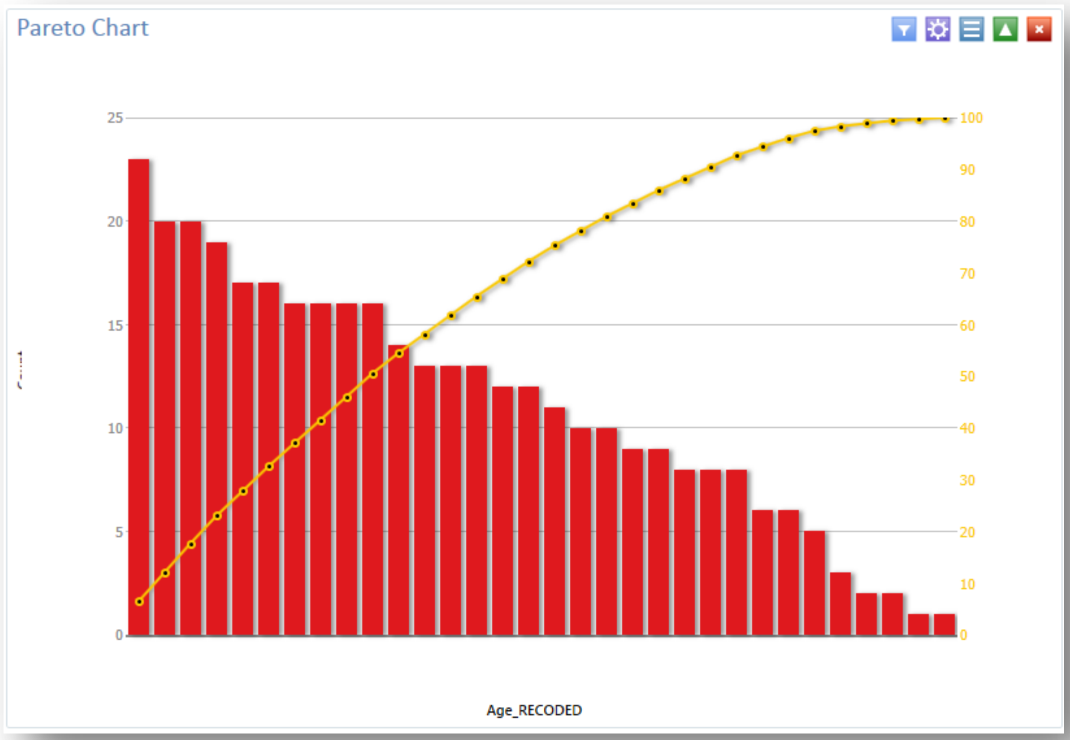


Figure 7.75: Pareto Chart

Using Advanced Statistics

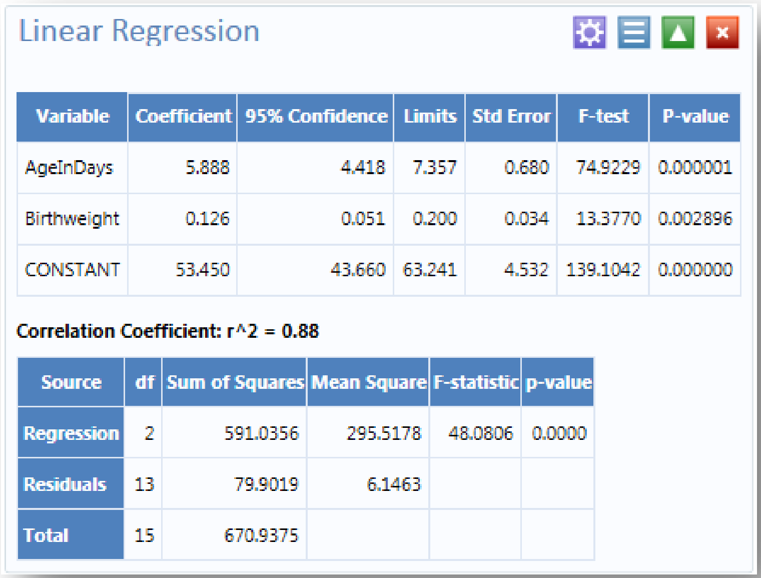
Linear Regression

Linear regression is an analysis tool that identifies a relationship between a continuous variable and one or more independent variables. It can be used for simple linear regression (only one independent variable), for multiple linear regression (more than one independent variable), and for quantifying the relationship between two continuous variables (correlation). Linear regression is used when you want to determine the relationship between one dependent variable with one or more independent variables.

In the example below, linear regression is used to determine if systolic blood pressure can be predicted by an infant's birth weight and age. The dependent variable is systolic blood pressure, and the independent variables are birth weight in ounces and age in days.

1. Select the **Sample.prj** project. Open the **BabyBloodPressure** form from the **Data Source Explorer** menu.

2. Click **OK.**
3. Right click on the canvas and select **Add Analysis Gadget** > **Advanced Statistics > Linear Regression**. The **Regression Properties** dialog box appears on the Visual Dashboard canvas.
4. From the **Outcome** Variable drop-down list, select **SystolicBlood**.
5. From the **Fields** drop-down list, select **AgeInDays** and **Birthweight**.
6. Click **Run.** Results appear in the Visual Dashboard canvas.


Linear Regression

Variable	Coefficient	95% Confidence	Limits	Std Error	F-test	P-value
AgeInDays	5.888	4.418	7.357	0.680	74.9229	0.000001
Birthweight	0.126	0.051	0.200	0.034	13.3770	0.002896
CONSTANT	53.450	43.660	63.241	4.532	139.1042	0.000000

Correlation Coefficient: r^2 = 0.88

Source	df	Sum of Squares	Mean Square	F-statistic	p-value
Regression	2	591.0356	295.5178	48.0806	0.0000
Residuals	13	79.9019	6.1463		
Total	15	670.9375			

Figure 7.76: Linear Regression Results

Both independent variables' coefficients are positive and the F-statistic associated with each independent variable is highly significant (p-value < 0.01), suggesting that each variable is a predictor of higher systolic blood pressure.

Linear Regression Properties

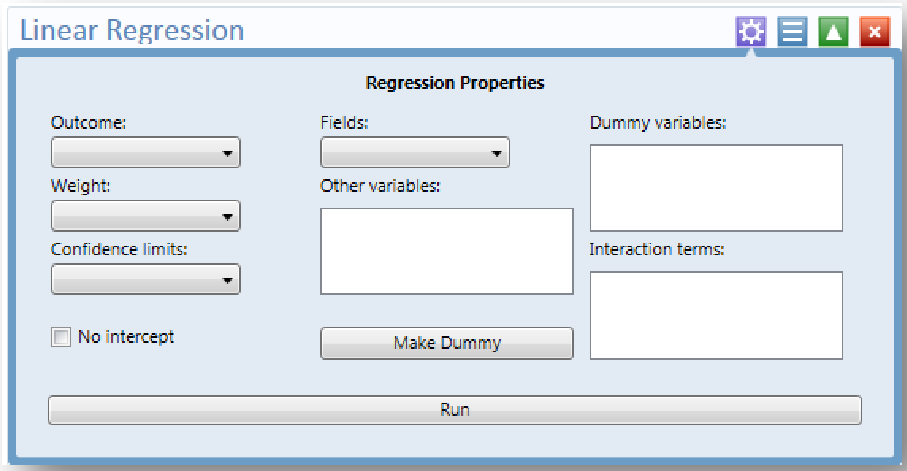


Figure 7.77: Linear Regression Properties

- The **Outcome** is the dependent variable for the regression. The outcome variable must be numeric or a Yes/No field.
- A **Weight** variable may be selected to use in weighted analyses.
- The **Confidence Limits** specifies the probability level at which confidence limits are computed.
- If **No Intercept** is selected, the regression is performed without a constant term, forcing the regression line through the origin.
- The **Fields** drop-down list contains the predictor (independent) variables.
- When selected, the predictor variables will appear in the **Other variables** list box. Double click on a variable to remove it from the **Other variables** list box.
- Select a variable from the **Other Variables** list box to activate the **Make Dummy** button. The selected variable will then appear in the **Dummy Variable** list box.
- **Interaction Terms** are defined with the **Make Interaction** button. Make Interaction appears if two or more variables are selected from the **Other Variables** list box. If you click **Make Interaction**, the relationship populates the **Interaction Terms** list box. Make Interaction adds all possible combinations of the selected variables to the regression as interaction terms.

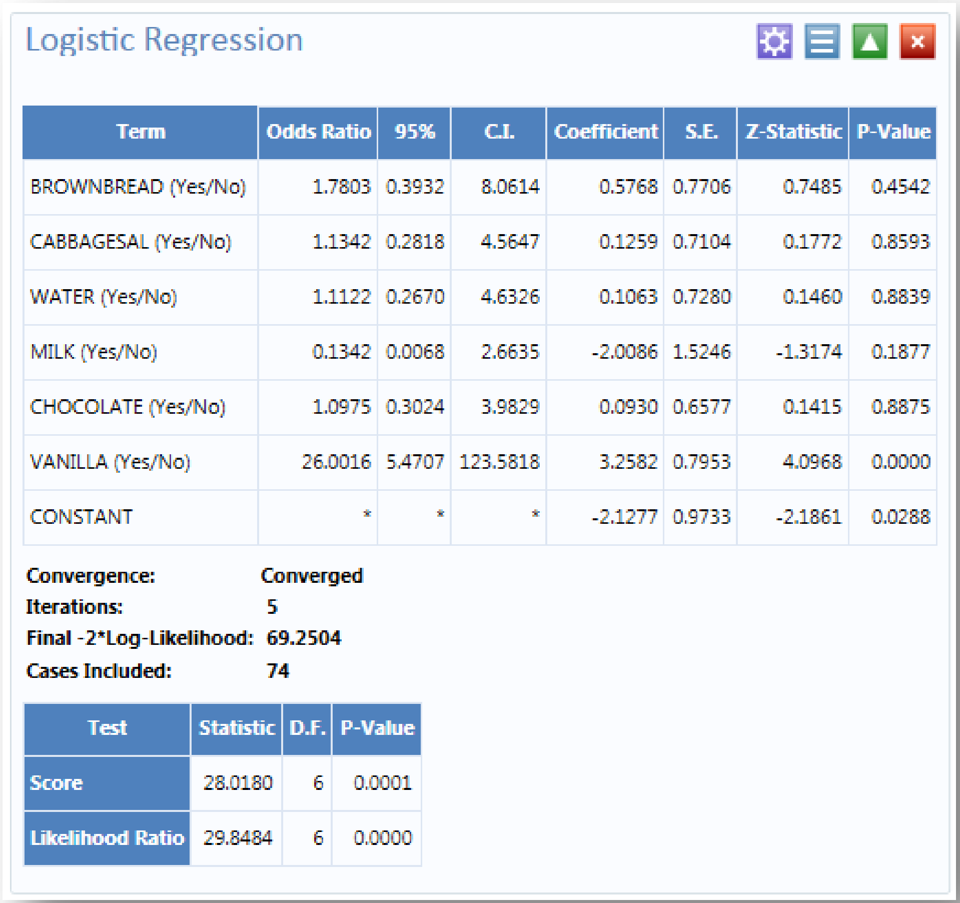
Logistic Regression

In Epi Info™ 7, either the M x N/ 2 x 2 table or Logistic Regression may be used when the outcome variable is dichotomous (for example, Ill- Yes/ Ill- No). However, an M x N/ 2 x 2 analysis is useful only when there is one "risk factor". Logistic regression is useful when the number of explanatory variables ("risk factors") is more than one. Logistic regression shows the relationship between an outcome variable with two values and explanatory variables that can be categorical or continuous. To use Logistic Regression, the dependent (outcome) variable must have a Yes/No value but independent (other variables) can be numeric, categorical, or Yes/No variables.

Records with missing values are excluded from the analyses. If **Include Missing** is used with missing values and Yes/No fields, dummy variables will generate automatically, which contribute Yes vs. Missing and No vs. Missing. Independent variables of text type are automatically turned into dummy variables, which compare each value relative to the lowest value in the sort order. Date or numeric type independent variables are treated as continuous variables unless they are set as dummy variables, which compare each value relative to the lowest value.

In the following example, logistic regression is used to determine the odds ratio of six foods that could be the cause of a hypothetical food borne illness. The dependent variable is Ill, and the independent variables are brownbread, cabbage, water, milk, chocolate, and vanilla.

1. Select the **Sample.PRJ** Data Source. Open the **Oswego** form from the **Data Source Explorer** menu.
2. Click **OK.**
3. Right click on the canvas and select **Add Analysis Gadget** > **Advanced Statistics > Logistic Regression**. The **Regression Properties** dialog box appears on the Visual Dashboard canvas.
4. From the **Outcome** drop-down list, select **ILL**.
5. From the **Fields** drop-down list, select **BROWNBREAD**, **CABBAGESAL**, **WATER**, **MILK**, **CHOCOLATE**, and **VANILLA**.
6. Click **Run**. Results appear on the Visual Dashboard canvas.



Term	Odds Ratio	95%	C.I.	Coefficient	S.E.	Z-Statistic	P-Value
BROWNBREAD (Yes/No)	1.7803	0.3932	8.0614	0.5768	0.7706	0.7485	0.4542
CABBAGESAL (Yes/No)	1.1342	0.2818	4.5647	0.1259	0.7104	0.1772	0.8593
WATER (Yes/No)	1.1122	0.2670	4.6326	0.1063	0.7280	0.1460	0.8839
MILK (Yes/No)	0.1342	0.0068	2.6635	-2.0086	1.5246	-1.3174	0.1877
CHOCOLATE (Yes/No)	1.0975	0.3024	3.9829	0.0930	0.6577	0.1415	0.8875
VANILLA (Yes/No)	26.0016	5.4707	123.5818	3.2582	0.7953	4.0968	0.0000
CONSTANT	*	*	*	-2.1277	0.9733	-2.1861	0.0288

Convergence:	Converged
Iterations:	5
Final -2*Log-Likelihood:	69.2504
Cases Included:	74

Test	Statistic	D.F.	P-Value
Score	28.0180	6	0.0001
Likelihood Ratio	29.8484	6	0.0000

Figure 7.78 Logistic Regression Results

The results show that Vanilla has an Odds Ratio and Confidence Interval significantly greater than one. This indicates that consumption of vanilla was likely the cause of food borne illness.

Logistic Regression Properties

- The **Outcome** variable is the dependent variable for the regression. The outcome variable must be numeric or a Yes/No field.
- A **Weight** variable may be selected to use in weighted analyses.
- **Match Variable** identifies the variable indicating the group membership of each record.
- **The Confidence Limits** specifies the probability level at which confidence limits are computed.
- If **No Intercept** is selected, the regression is performed without a constant term.

- The **Include Missing** setting controls independent variables. If **Include Missing** is used with missing values and true/false, dummy variables will be made automatically. This will contribute Yes vs. Missing and No vs. Missing. Independent variables of text type are automatically turned into dummy variables, which compare each value relative to the lowest value in the sort order. Date or numeric type Independent variables are treated as continuous variables unless designated as dummy values through the **Make Dummy** button. If that occurs, they automatically turn into dummy variables, which compare each value relative to the lowest value.
- The **Fields** drop-down list contains the predictor (independent) variables.
- When selected, the predictor variables will appear in the **Other variables** list box. Double click on a variable to remove it from the **Other** variables list box. Select a variable from the **Other variables** list box to activate the **Make Dummy** button. The selected variable will then appear in the **Dummy variable** list box. Double click on a variable to remove it from the **Dummy variable** list box,
- **Interaction terms** are defined with the **Make Interaction** button. Make Interaction appears if two or more variables are selected from the Other Variables list box. If you click **Make Interaction**, the relationship populates the Interaction terms list box. Make Interaction adds all possible combinations of the selected variables to the regression as interaction terms. Double click on a variable to remove it from the **Interaction terms** list box.
- The Clear Terms button clears all variables from the Dummy variables and Interaction Terms list boxes.

Complex Sample Frequencies, Tables, and Means

The Frequency, Table, and Means options in Epi Info™ 7 perform statistical calculations assuming the data were collected using simple random sampling (SRS) or unbiased systematic sampling. More complicated sampling strategies such as stratification, cluster sampling, and the use of unequal sampling fractions are used in many surveys. Visual Dashboard provides three options to analyze complex sample data: Complex Sample Frequencies, Complex Sample Means, and Complex Sample Tables.

Generally, in complex sample analysis, there is a variable for the primary sampling unit (PSU) or Cluster from which a sample subject was selected. If the PSUs were chosen from different Strata (e.g., states or counties), there may be a stratification variable (Stratify by). The concept of sample stratification in complex sample design differs from the concept of stratification during epidemiologic analysis using the TABLES command, as the Strata are chosen in the sampling process before analysis. In addition, a weight variable (Weight) is used when sampling strategies result in unequal selection probabilities. The complex sample commands in Epi Info™ 7 can compute proportions or means with standard errors and confidence limits. If a 2x2 table is requested, the odds ratio, risk ratio, and risk difference are provided.

Complex Sample Frequencies

The Complex Samples Frequencies procedure produces frequency tables for selected variables. The Sample project contains an Expanded Program for Immunization (EPI) cluster survey. Using the EPI method, a team selected 30 communities (i.e., clusters) from the chosen geographic area and visited each of the 30 communities. In each, they selected seven children in an appropriate age range and determined each child's immunization (VAC) status. The following example will determine the frequency of vaccinations using Complex Sample Frequency.

1. Open the **Sample.PRJ** project.
2. Select the **Epi1** form from the **Data Source Explorer** menu.
3. Right click on the canvas to display the menu options.
4. Select **Add Analysis Gadget > Advanced Statistics > Complex Sample Frequencies**.
5. The **Complex Sample Frequency Properties** dialog box appears.

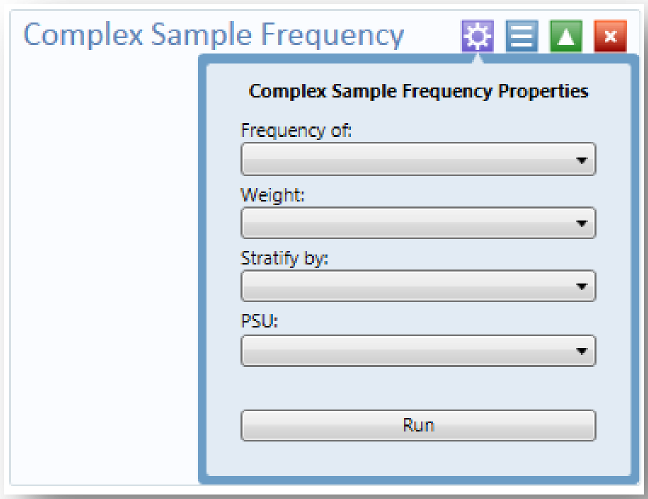
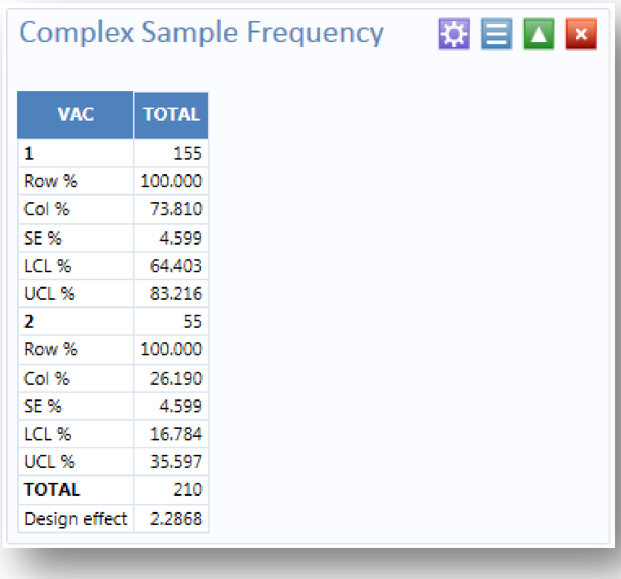


Figure 7.79: Complex Sample Frequency Properties

Complex Sample Frequency Options:

- **Frequency of** identifies the variable(s) whose frequency is computed.
- A **Weight** variable is selected for use in weighted analyses.
- **Stratify by** identifies the variable to be used to stratify or group the frequency data.
- The **PSU** identifies the Primary Sampling Unit.

6. From the **Frequency Of** drop-down list, select **VAC**.
7. From the **PSU** drop-down list, select **Cluster.**
8. Click **Run**.
9. The **Complex Sample Frequency** results appear on the canvas.



VAC	TOTAL
1	155
Row %	100.000
Col %	73.810
SE %	4.599
LCL %	64.403
UCL %	83.216
2	55
Row %	100.000
Col %	26.190
SE %	4.599
LCL %	16.784
UCL %	35.597
TOTAL	210
Design effect	2.2868

Figure 7.80: Complex Sample Frequency Results

Information provided in the output includes:

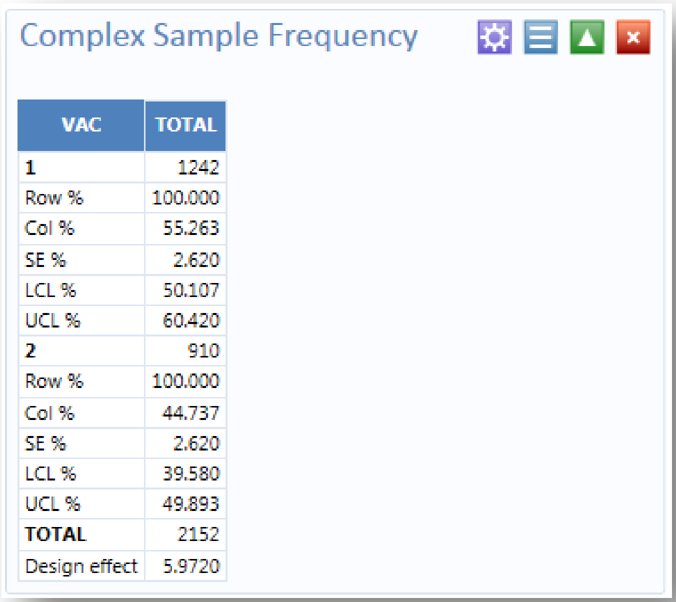
- **Row %** - the row percentage, frequency will always be 100%.
- **Col %** - the column percentage.
- **SE %** - the standard error, which takes into account the complex sample design.
- **LCL %** - Lower Confidence Limit.
- **UCL %** - Upper Confidence Limit.
- **Total** - total number of individuals/elements surveyed.

The results indicate that 73.8% of the 210 children surveyed are vaccinated with a 95% confidence interval range from 64.4 to 83.2.

The following example is similar to the one above, except that this is a stratified cluster survey with a separate 30-cluster survey completed in each of 10 strata. To analyze this dataset correctly, we will take into account where each child lives (LOCATION). We will also use a weight variable to account for the differences in population sizes between the different locations.

1. Open the **Sample.PRJ** project.
2. Select the **Epi10** form from the **Data Source Explorer** menu.
3. Right click on the canvas to display the menu options.
4. Select **Add Analysis Gadget > Advanced Statistics > Complex Sample Frequencies**.
5. The **Complex Sample Frequency Properties** dialog box appears.
6. From the **Frequency Of** drop-down list, select **VAC**.

7. From the **Weight** drop-down list, select **POPW**.
8. From the **Stratify By** drop-down list, select **Location**.
9. From the **PSU** drop-down list, select **Cluster.**
10. Click **Run**.
11. The **Complex Sample Frequency** results appear on the canvas.



Complex Sample Frequency

VAC	TOTAL
1	1242
Row %	100.000
Col %	55.263
SE %	2.620
LCL %	50.107
UCL %	60.420
2	910
Row %	100.000
Col %	44.737
SE %	2.620
LCL %	39.580
UCL %	49.893
TOTAL	2152
Design effect	5.9720

Figure 7.81: Complex Sample Frequency (Stratified, Weighted) Results

The results indicate that 55.3% of the 2,152 children surveyed are vaccinated with a 95% confidence interval range from 50.1to 60.4.

Complex Sample Means

The Complex Sample Means command can be used when the outcome variable is continuous, such as age, cholesterol level, etc. You can either calculate an overall mean with its measures of variation or compare means across a grouping variable.

As an example of calculating means with a grouping variable, use the Smoke data file located in the Sample project folder. In this example, the investigator is interested in determining if, among smokers, there is a difference in the average number of cigarettes smoked between males and females. In these data, the variable SEX is coded as 1=male and 2=female.

1. Open the **Sample.PRJ** project.
2. Select the **Smoke** form from the **Data Source Explorer** menu.
3. Right click on the canvas to display the menu options.
4. Select **Add Analysis Gadget > Advanced Statistics > Complex Sample Means**.

5. The **Complex Sample Frequency Means** dialog box appears.

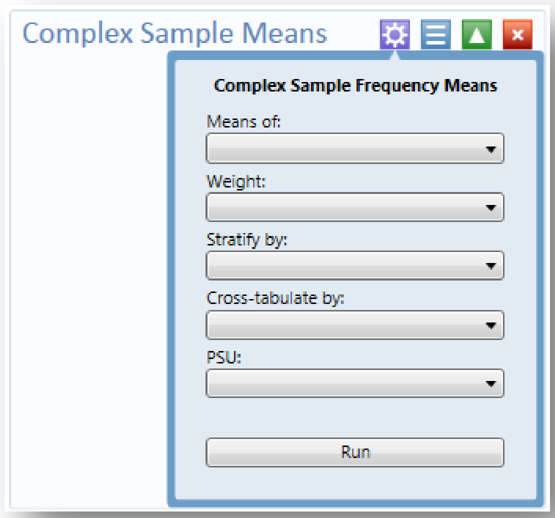
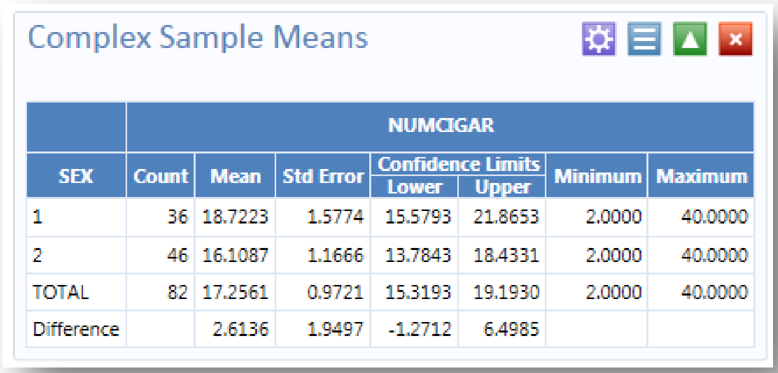


Figure 7.82: Complex Sample Frequency Means

Complex Sample Tables Options:

- **Means of** identifies the variable whose mean is to be computed.
- A **Weight** variable is selected for use in weighted analyses.
- **Stratify by** identifies the variable to be used to stratify or group the frequency data.
- **Cross-Tabulate by** identifies the variable to be used to cross-tabulate the main variable.
- The **PSU** identifies the Primary Sampling Unit.

6. From the **Means Of** drop-down list, select **Numcigar**.
7. From the **Weight** drop-down list, select **SAMPW**.
8. From the **Stratify By** drop-down list, select **Strata**.
9. From the **Cross-tabulate By** drop-down list, select **Sex.**
10. From the **PSU** drop-down list, select **PSUID.**
11. Click **Run**.
12. The **Complex Sample Means** results appear on the canvas.


Complex Sample Means

	NUMCIGAR						
SEX	Count	Mean	Std Error	Confidence Limits Lower	Confidence Limits Upper	Minimum	Maximum
1	36	18.7223	1.5774	15.5793	21.8653	2.0000	40.0000
2	46	16.1087	1.1666	13.7843	18.4331	2.0000	40.0000
TOTAL	82	17.2561	0.9721	15.3193	19.1930	2.0000	40.0000
Difference		2.6136	1.9497	-1.2712	6.4985		

Figure 7.83: Complex Sample Means Results

The results indicate that among the 82 individuals who smoked cigarettes, the average number of cigarettes smoked per day for men was 18.7, and 16.1 for women with a 95% confidence interval range of 15.6 to 21.8 for men and 13.7 to 18.4 for women. Note that the Smoke file has 337 individuals. However, the number of cigarettes smoked per day (NUMCIGAR) has data for only the 82 smokers. For nonsmokers this variable was left blank and therefore is treated as missing data and excluded from analysis.

Complex Sample Tables

The Complex Sample Tables option in Visual Dashboard allows you to specify an Exposure Variable and an Outcome Variable. The following example using Complex Sample Tables to show whether the mother received prenatal care (PRENATAL) has an effect on the child's vaccination status. If the mother had received prenatal care, PRENATAL=1 else PRENATAL=2.

1. Open the **Sample.PRJ** project.
2. Select the **Epi10** form from the **Data Source Explorer** menu.
3. Right click on the canvas to display the menu options.
4. Select **Add Analysis Gadget > Advanced Statistics > Complex Sample Tables**.
5. The **Complex Sample Tables Properties** dialog box appears.

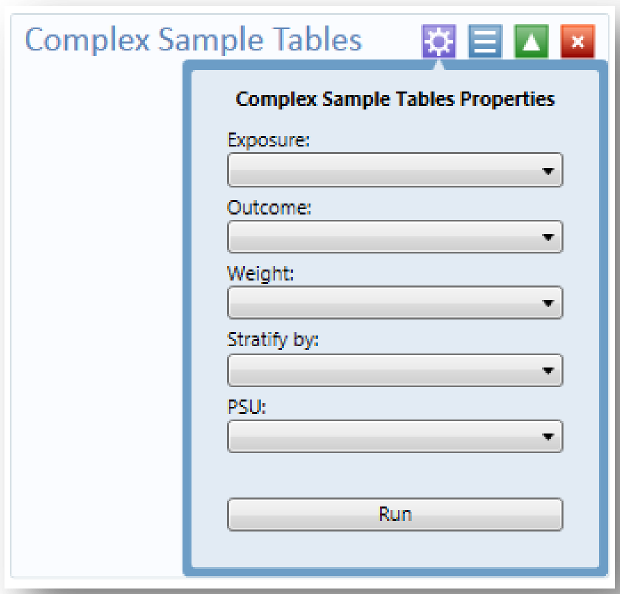
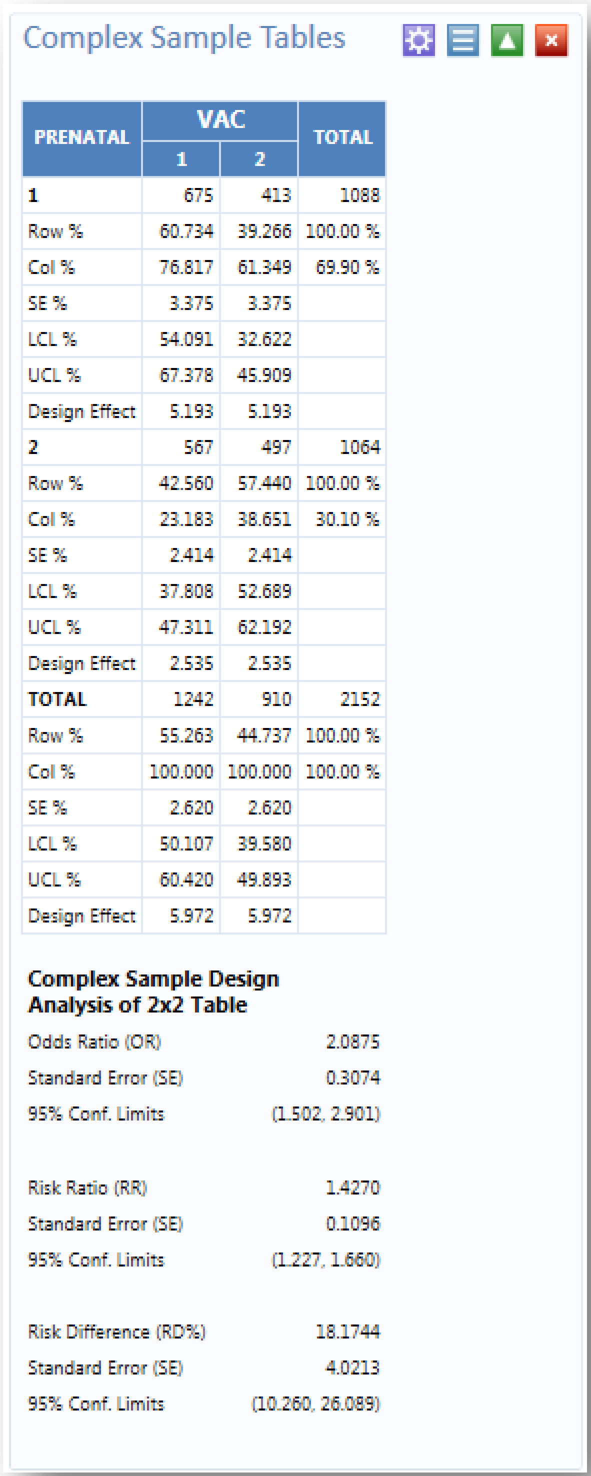


Figure 7.84: Complex Sample Tables Properties

Complex Sample Tables Options:

- **Exposure Variable** identifies the variable that will appear on the horizontal axis of the table. It is considered to be the risk factor (or * for all variables).
- **Outcome Variable** identifies the variable that will appear on the vertical axis of the table.
- A **Weight** variable is selected for use in weighted analyses.
- **Stratify by** identifies the variable to be used to stratify or group the frequency data.
- The **PSU** identifies the Primary Sampling Unit.

6. From the **Exposure** drop-down list, select **Prenatal**.
7. From the **Outcome** drop-down list, select **VAC**.
8. From the **Weight** drop-down list, select **POPW**.
9. From the **Stratify By** drop-down list, select **Location.**
10. From the **PSU** drop-down list, select **Cluster.**
11. Click **Run**.
12. The **Complex Sample Tables** results appear on the canvas.



Complex Sample Tables

PRENATAL	VAC		TOTAL
	1	2	
1	675	413	1088
Row %	60.734	39.266	100.00 %
Col %	76.817	61.349	69.90 %
SE %	3.375	3.375	
LCL %	54.091	32.622	
UCL %	67.378	45.909	
Design Effect	5.193	5.193	
2	567	497	1064
Row %	42.560	57.440	100.00 %
Col %	23.183	38.651	30.10 %
SE %	2.414	2.414	
LCL %	37.808	52.689	
UCL %	47.311	62.192	
Design Effect	2.535	2.535	
TOTAL	1242	910	2152
Row %	55.263	44.737	100.00 %
Col %	100.000	100.000	100.00 %
SE %	2.620	2.620	
LCL %	50.107	39.580	
UCL %	60.420	49.893	
Design Effect	5.972	5.972	

Complex Sample Design Analysis of 2x2 Table

Odds Ratio (OR)	2.0875
Standard Error (SE)	0.3074
95% Conf. Limits	(1.502, 2.901)
Risk Ratio (RR)	1.4270
Standard Error (SE)	0.1096
95% Conf. Limits	(1.227, 1.660)
Risk Difference (RD%)	18.1744
Standard Error (SE)	4.0213
95% Conf. Limits	(10.260, 26.089)

Figure 7.85: Complex Sample Tables Results

The results show that 60.7% of children whose mothers received prenatal care were immunized compared to 42.7% of those children whose mothers did not receive prenatal care.

The 2 x 2 data shows that the odds ratio in the data was 2.088, the risk ratio was 1.427 and this risk difference is 18.2%. The prevalence ratio says that 1.427 times as many children of women who received prenatal care were immunized (60.734% /42.560% = 1.427) compared to children born to women who had not received prenatal care, a 40% difference.

StatCalc Calculator

Refer to the StatCalc Calculator section of this user guide.

NutStat Growth Chart

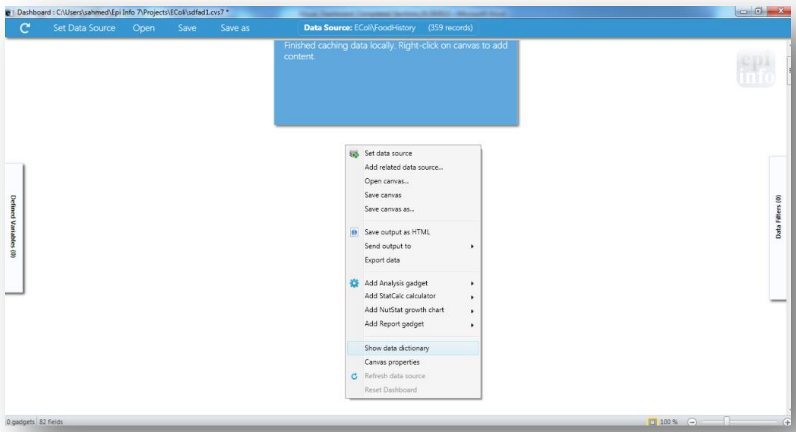
Refer to the NutStat section of this user guide.

Data Dictionary

Show Data Dictionary

The Data Dictionary displays form(s) and defined variables for an open project. Fields or variables are sorted and displayed by page number in the form. Defined variables appear at the end of the listing. Information retrieved from the form includes Field Name, Prompt, Form, Page, Tab, Data Type, Epi Info Field Type, and Table.

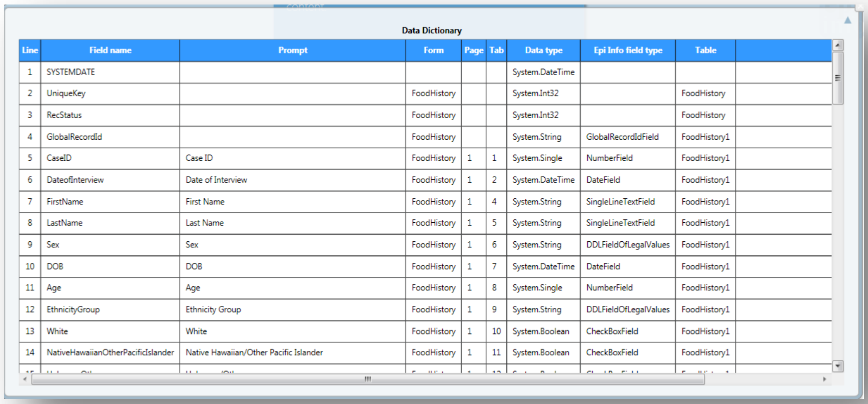
1. Right click on the canvas and select **Show Data Dictionary**.



Finished caching data locally. Right-click on canvas to add content.

Figure 7.86: Data Dictionary Menu Option

2. The **Data Dictionary** is displayed in **Visual Dashboard**.



Data Dictionary

Line	Field name	Prompt	Form	Page	Tab	Data type	Epi Info field type	Table
1	SYSTEMDATE					System.DateTime		
2	UniqueKey		FoodHistory			System.Int32		FoodHistory
3	RecStatus		FoodHistory			System.Int32		FoodHistory
4	GlobalRecordId		FoodHistory			System.String	GlobalRecordIdField	FoodHistory1
5	CaseID	Case ID	FoodHistory	1	1	System.Single	NumberField	FoodHistory1
6	DateofInterview	Date of Interview	FoodHistory	1	2	System.DateTime	DateField	FoodHistory1
7	FirstName	First Name	FoodHistory	1	4	System.String	SingleLineTextField	FoodHistory1
8	LastName	Last Name	FoodHistory	1	5	System.String	SingleLineTextField	FoodHistory1
9	Sex	Sex	FoodHistory	1	6	System.String	DDLFieldOfLegalValues	FoodHistory1
10	DOB	DOB	FoodHistory	1	7	System.DateTime	DateField	FoodHistory1
11	Age	Age	FoodHistory	1	8	System.Single	NumberField	FoodHistory1
12	EthnicityGroup	Ethnicity Group	FoodHistory	1	9	System.String	DDLFieldOfLegalValues	FoodHistory1
13	White	White	FoodHistory	1	10	System.Boolean	CheckBoxField	FoodHistory1
14	NativeHawaiianOtherPacificIslander	Native Hawaiian/Other Pacific Islander	FoodHistory	1	11	System.Boolean	CheckBoxField	FoodHistory1

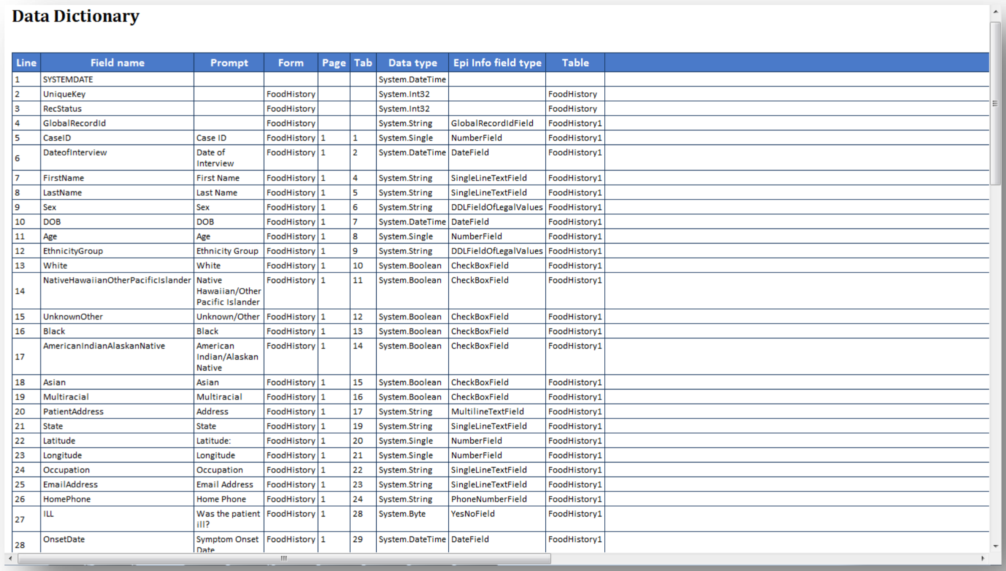
Figure 7.87: Data Dictionary View

Column values for **Prompt**, **Data Type**, **Field Name** and **Epi Info Field type** are developed when fields are created from the **Field Definition** dialog box.

1. To open the **Data Dictionary** as an **HTML** page inside the browser window, right click and then select **Send Data Dictionary to Web Browser**.

From the browser, the data can be printed with **File > Print,** or saved with **File > Save As**.

4. Right click on the **HTML page** to show the pop-up menu. You can export the data directly to an Excel spreadsheet.



Data Dictionary

Line	Field name	Prompt	Form	Page	Tab	Data type	Epi Info field type	Table
1	SYSTEMDATE					System.DateTime		
2	UniqueKey		FoodHistory			System.Int32		FoodHistory
3	RecStatus		FoodHistory			System.Int32		FoodHistory
4	GlobalRecordId		FoodHistory			System.String	GlobalRecordIdField	FoodHistory1
5	CaseID	Case ID	FoodHistory	1	1	System.Single	NumberField	FoodHistory1
6	DateofInterview	Date of Interview	FoodHistory	1	2	System.DateTime	DateField	FoodHistory1
7	FirstName	First Name	FoodHistory	1	4	System.String	SingleLineTextField	FoodHistory1
8	LastName	Last Name	FoodHistory	1	5	System.String	SingleLineTextField	FoodHistory1
9	Sex	Sex	FoodHistory	1	6	System.String	DDLFieldOfLegalValues	FoodHistory1
10	DOB	DOB	FoodHistory	1	7	System.DateTime	DateField	FoodHistory1
11	Age	Age	FoodHistory	1	8	System.Single	NumberField	FoodHistory1
12	EthnicityGroup	Ethnicity Group	FoodHistory	1	9	System.String	DDLFieldOfLegalValues	FoodHistory1
13	White	White	FoodHistory	1	10	System.Boolean	CheckBoxField	FoodHistory1
14	NativeHawaiianOtherPacificIslander	Native Hawaiian/Other Pacific Islander	FoodHistory	1	11	System.Boolean	CheckBoxField	FoodHistory1
15	UnknownOther	Unknown/Other	FoodHistory	1	12	System.Boolean	CheckBoxField	FoodHistory1
16	Black	Black	FoodHistory	1	13	System.Boolean	CheckBoxField	FoodHistory1
17	AmericanIndianAlaskanNative	American Indian/Alaskan Native	FoodHistory	1	14	System.Boolean	CheckBoxField	FoodHistory1
18	Asian	Asian	FoodHistory	1	15	System.Boolean	CheckBoxField	FoodHistory1
19	Multiracial	Multiracial	FoodHistory	1	16	System.Boolean	CheckBoxField	FoodHistory1
20	PatientAddress	Address	FoodHistory	1	17	System.String	MultilineTextField	FoodHistory1
21	State	State	FoodHistory	1	19	System.String	SingleLineTextField	FoodHistory1
22	Latitude	Latitude:	FoodHistory	1	20	System.Single	NumberField	FoodHistory1
23	Longitude	Longitude	FoodHistory	1	21	System.Single	NumberField	FoodHistory1
24	Occupation	Occupation	FoodHistory	1	22	System.String	SingleLineTextField	FoodHistory1
25	EmailAddress	Email Address	FoodHistory	1	23	System.String	SingleLineTextField	FoodHistory1
26	HomePhone	Home Phone	FoodHistory	1	24	System.String	PhoneNumberField	FoodHistory1
27	ILL	Was the patient ill?	FoodHistory	1	28	System.Byte	YesNoField	FoodHistory1
28	OnsetDate	Symptom Onset Date	FoodHistory	1	29	System.DateTime	DateField	FoodHistory1

Figure 7.88: Data Dictionary Export to HTML

3. Click **Close** to exit the **Data Dictionary**.

Note: The Data Dictionary can be refreshed at any time to display the most current data. Right click on the Data Dictionary and select Refresh Data Dictionary.

Report Gadgets

There are several report gadgets available to add text and images to your reports. The **Simple Text box** option allows you to specify several lines of text to appear at the top of the table, frequency, chart, or graph. This is particularly useful for adding a title to charts or graphs. To add a title to your graph:

1. Right click on the **Visual Dashboard canvas**. Select **Add Report Gadget** from the menu options, and then select **Simple Text box**.
2. The text box appears.
3. Enter a title for your graph.

To remove your title:

1. Right click on the **textbox** border.
2. Click **Close this gadget**.

Image Gadget is another report gadget, which allows you to add an image to your charts or graphs.

1. Right click on the **Visual Dashboard canvas**. Select **Add Report Gadget** from the menu options, and then select **Image Gadget**.
2. A light blue box appears on the canvas. Click the **box**.
3. The **Open** dialog box appears. Select the location of your image or enter the **File name** of your image.

Note: You may only select a * .png file.

4. Click **Open**.
5. The image appears on the canvas.

To remove your image:

1. Right click on your image.
2. Click **Close this gadget**.

THIS PAGE IS

INTENTIONALLY BLANK.

9. Classic Analysis

Introduction

Classic Analysis manipulates, manages, and analyzes data. It acts as a statistical toolbox providing many ways to transform data and perform statistical Classic Analysis. Data can be selected, sorted, listed, or manipulated with a series of commands, functions, and operators. Available statistics include frequencies, means, and more advanced processes (i.e., Kaplan-Meier Survival Classic Analysis and Logistic Regression). Classic Analysis can be accessed by clicking **Classic** from the Epi Info™ main window or by selecting **Tools > Analyze Data>Classic** . It reads data files created in Form Designer and other types of databases (e.g., MS Access, MS Excel, SQL Server, and ASCII). Classic Analysis can also produce graphs to present graphic representations of data.

Classic Analysis provides access to existing data directly or through forms created in Form Designer. It reads data from files and tables created in Epi Info 7, Microsoft Access, Microsoft Excel, SQL Server, and ASCII.

Navigate the Classic Analysis Workspace

The Classic Analysis module contains four areas: the Classic Analysis Command Tree, Program Editor, Classic Analysis Output window, and the Message Area.

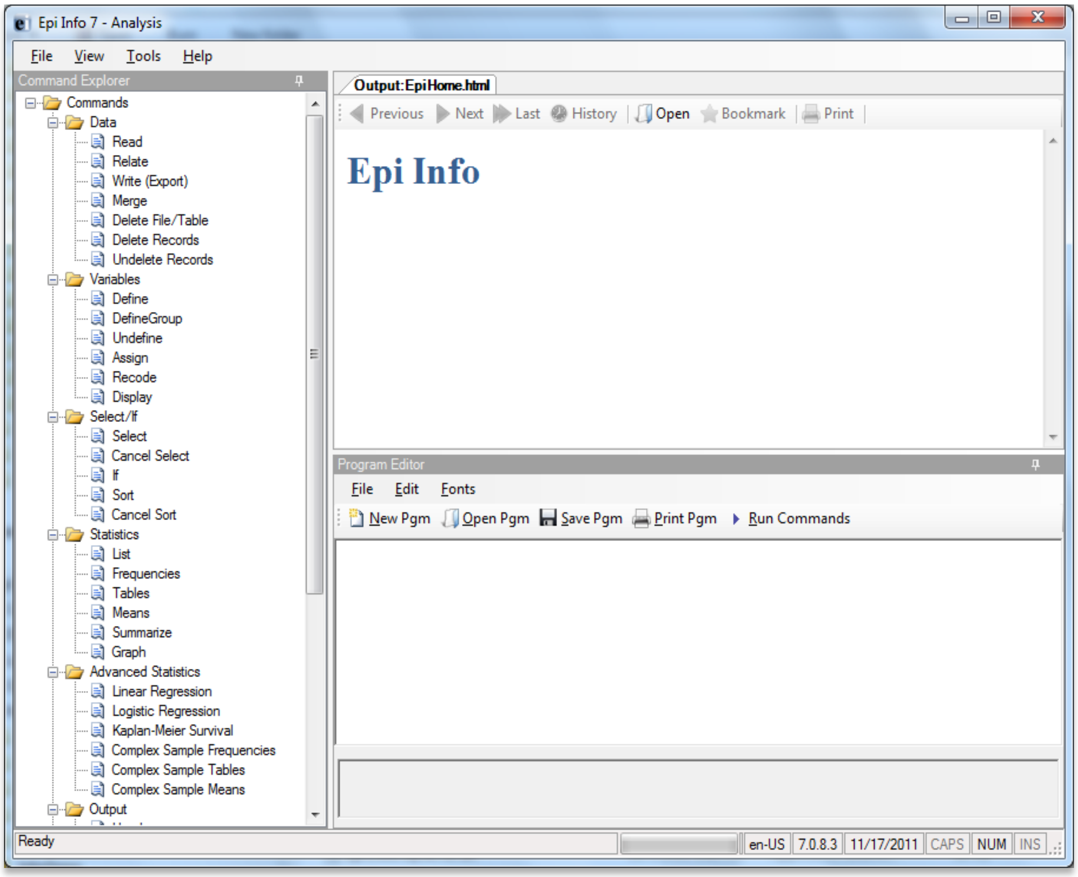


Figure 7.89: Classic Analysis Workspace

1. The Command Explorer contains a list of available commands separated into folders by command type. Commands are generated, edited, and executed. Selecting a command opens the corresponding dialog box for that command, function, or statistic to run.

2. The Program Editor displays the commands and code created using the Classic Analysis Command Tree. Commands can also be typed directly into the Program Editor. Programs or .PGM files written in Classic Analysis can be stored in the current .PRJ or as text files. Saved programs can be run against new data, or shared with others.

3. The Output window acts as a browser and displays information generated from commands run in Program Editor. The buttons allow you to navigate through program scripts that run and display in the output screen.

4. The Message window alerts you if any problems occur with any executed commands.

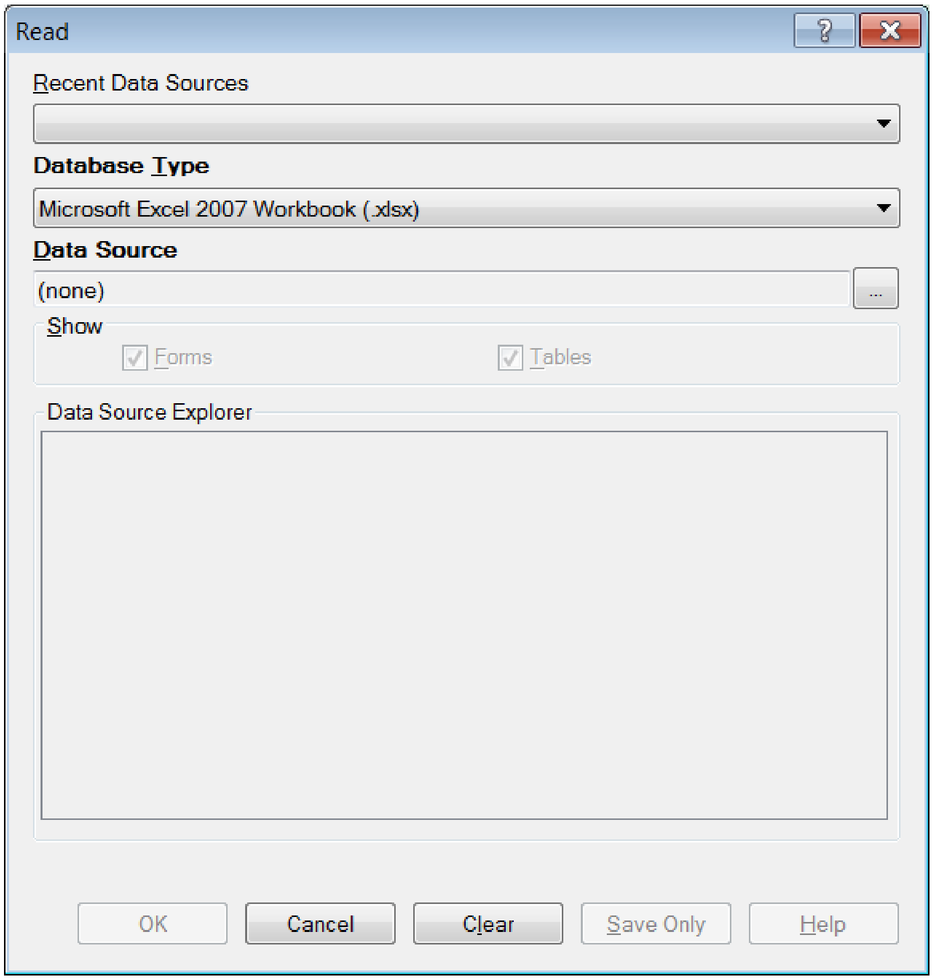
How to Manage Data

Use the READ Command

To analyze data, it must be read or imported into Classic Analysis. The READ command allows you to select a project and/or data table to run statistics. The READ command is used almost every time you open Classic Analysis.

Warning! Never manipulate the Form table in a software application outside of Epi Info 7. The database may become corrupt and render the form unusable.

1. From the Classic Analysis Command Tree Data folder, click **Read.** The READ dialog box opens.



Database Type: Microsoft Excel 2007 Workbook (.xlsx)

Data Source: (none)

Figure 7.90: Read Window

- The **Database Type** field indicates the database file to be loaded (.PRJ, .MDB, .XLS). Specify the data format for the data to be read. Unless otherwise specified, output files created from these data are stored in this destination folder.
- The **Data Source** field indicates the file location/path. If you use an SQL Server database, the Data Source field will require server and database names.

2. The **Recent Data Sources** field provides a list of recently-accessed databases in Classic Analysis or Visual Dashboard. By selecting one of the data sources available on the list, you do not need to provide Database Type or Data Source information. Only data tables or forms will need to be selected again. If you are reading an Epi Info 7 project file from the Data Source section, select the **Forms** checkbox to view available forms in the project, or select the **Tables** checkbox to view all project tables.
3. From the Forms section, select a **form** in your project. This reads the **data table** associated with your form.
4. Click **OK**. The READ command is saved in the Program Editor and run simultaneously. The current form file location, Record Count, and Date appear in the Classic Analysis Output window; the READ command appears in the Program Editor.
    - Click **Save Only** to save the command in the Program Editor. The command does not run and the Record Count is not displayed.

Read a Microsoft Excel File

8. From the Classic Analysis Command Tree, click **Read.** The READ dialog box opens.
9. From the Data Formats drop-down list, select either **MS Excel 97-2003 Workbook** or **MS Excel 2007 Workbook.**
10. In the Data Source field, click on the **Browse** button to browse and select the **workbook (.xls or .xlsx)** to import into Classic Analysis.
11. Click **Open.**
    - First Row Contains Field Names is checked by default. If the first row of the spreadsheet does not contain field names, deselect the **checkbox**.
12. Click **OK**
13. Select a **Worksheet** from the list provided in the Data Source Explorer.
14. Click **OK**. The Record Count and file information appear in the Classic Analysis Output window.

Read a Microsoft Access File

8. From the Classic Analysis Command Tree, click **Read.** The READ dialog box opens.
9. From the Data Formats drop-down list, select either **MS Access 2002-2003 (.mdb)** or **MS Access 2007(.accdb).**
10. In the Data Source field, click on the **Browse** button to browse and select the **file** to import into Classic Analysis.
11. Click **OK.**
12. Select a **data table** from the list provided in the Data Source Explorer.
13. Click **OK**. The Record Count and file information appear in the Classic Analysis Output window.

Read an ASCII Text File

8. From the Classic Analysis Command Tree, click **Read**. The READ dialog box opens.
9. From the Data Formats drop-down list, select **Flat ASCII File**.
10. In the Data Source field, click the **Browse** button to browse.
11. Click the **Browse** button again. Select **directory** of the file to import into Classic Analysis.
12. Click **OK**.
13. Select the **file** from the list provided in the Data Source Explorer
14. Click **OK.**

Note: There are two forms of text files. Since both have only one table per file, you do not have to specify a table. Both put the data for one record on a single line. The difference is in how the fields are indicated.

Read an SQL Server Database

1. From the Classic Analysis Command Tree, click **Read**. The READ dialog box opens.
2. From the Data Formats drop-down list, select **Microsoft SQL Server Database**.
3. In the Data Source field, click the **Browse** button to browse.
4. When the Connect to SQL Server Database dialog opens, specify the server name and database name to connect and import into Classic Analysis.

5. Click **OK**.
6. From the list provided in the Data Source Explorer, select a **data table**.
7. Click **OK**. The Record Count and file information appear in the Classic Analysis Output window.

Use Related Forms

To use RELATE, the READ command must open at least one form/table. The form/table to be linked requires a key field that relates records in the two forms/tables and can be used to join them together. The keys in the main and related tables or forms do not require the same name. If the table was created in Form Designer and the data entry completed using Enter, Classic Analysis can establish a relationship automatically using the Global Record ID and Foreign Key variables created by Epi Info 7. If the relationship was created in another program that uses different keys, the key variables in both files must be identified. Relationships are represented by double colons (::). Classic Analysis can establish relationships using multiple keys. Currently, only Epi Info 7 projects can be related in Classic Analysis.

After issuing the RELATE command, the variables in the related table may be used as if they were part of the main table. Variable names are duplicated in the related tables; variable names will be suffixed with a sequence number. Frequencies, cross-tabulations, and other operations involving data in the main and related tables can be performed. The WRITE command can create a new table containing both sets of data. More than one table can be related to the main table by using a series of RELATE commands.

Relate keys can contain mathematical or string concatenation expressions. Epi Info 7 functions can be used to determine relationships. It may be easier, however, to write out a new file in which the complex key is included as a single variable. Use this single variable as a key.

PRJ File RELATE Syntax

Use READ to open the first PRJ file, and RELATE to open the second and create the relationship.

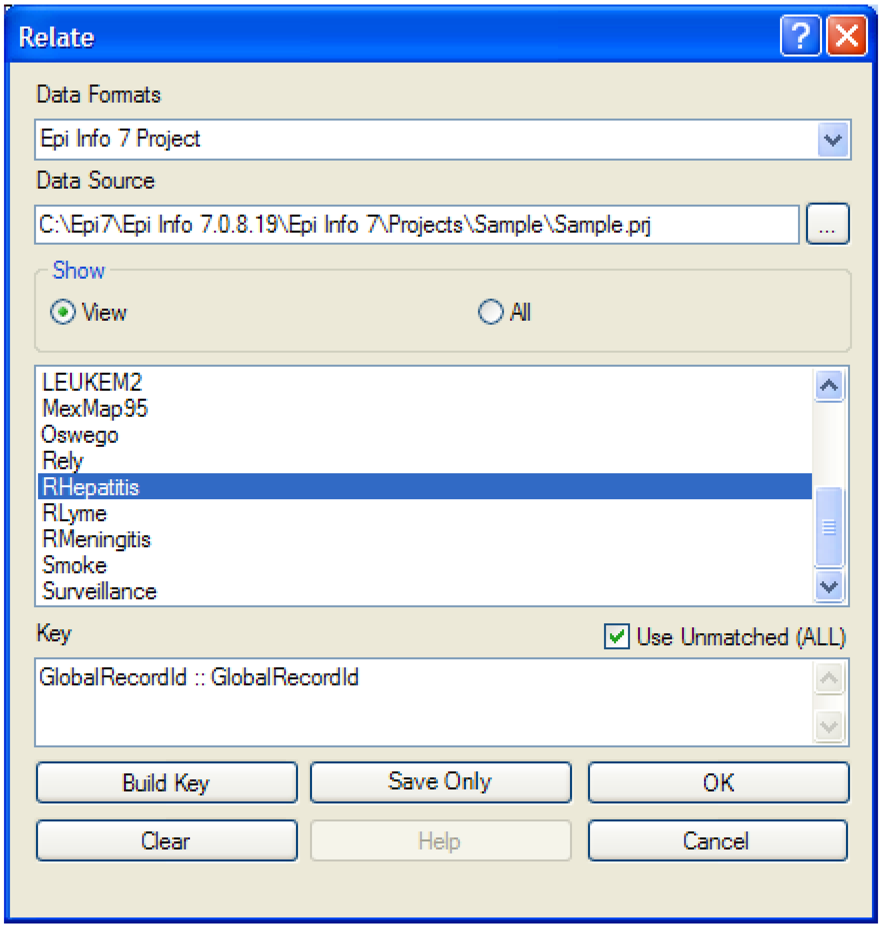
```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Surveillance

RELATE RHepatitis GlobalRecordId :: GlobalRecordId
```

Try It

Epi Info 7 has a related database in the Sample.PRJ file which contains two forms: Surveillance and RHepatitis. The variable GLOBAL RECORD ID is the internal Epi Info 7 identification key located in more than one form in the project.

5. Read in the **Sample.PRJ** project. Open **Surveillance**.
6. Click **RELATE**. The RELATE dialog box opens.

7. Select **RHepatitis**. The Build Key button activates.
8. Click **Build Key**. The RELATE-BUILD KEY dialog box opens. Make sure the **Current Table** radio button is selected.
7. From the Available Variables drop-down list, select **GLOBALRECORDID**.
8. Select the **Related Table** radio button.
9. From the Available Variables drop-down list, select **GLOBALRECORDID**.
10. Click **OK**. The Related Tables field populates.
11. Click **OK**. The RELATE dialog box opens and the Key field populates with the join information.


Relate

Data Formats: Epi Info 7 Project

Data Source: C:\Epi7\Epi Info 7.0.8.19\Epi Info 7\Projects\Sample\Sample.prj

Show: [✓ View] All

LEUKEM2
MexMap95
Oswego
Rely
RHepatitis
RLyme
RMeningitis
Smoke
Surveillance

Key: GlobalRecordId :: GlobalRecordId

[✓ Use Unmatched (ALL)]

Figure 7.91: Relate Dialog Box

7. Click **OK**. The related form information appears in the Output window.
   - Data from the two tables can now be used to compute statistics.

Use the WRITE Command

Use to create a new file with selected variables.

Syntax

```
WRITE <METHOD> {<output type>} {<project>:}table {[<variable(s)>]}

WRITE <METHOD> {<output type>} {<project>:}table * EXCEPT {[<variable(s)>]}
```

To use the WRITE command, open a **project** using the READ command. Follow these steps:

1. From the Classic Analysis Command Tree, click **Write**. The WRITE dialog box opens.

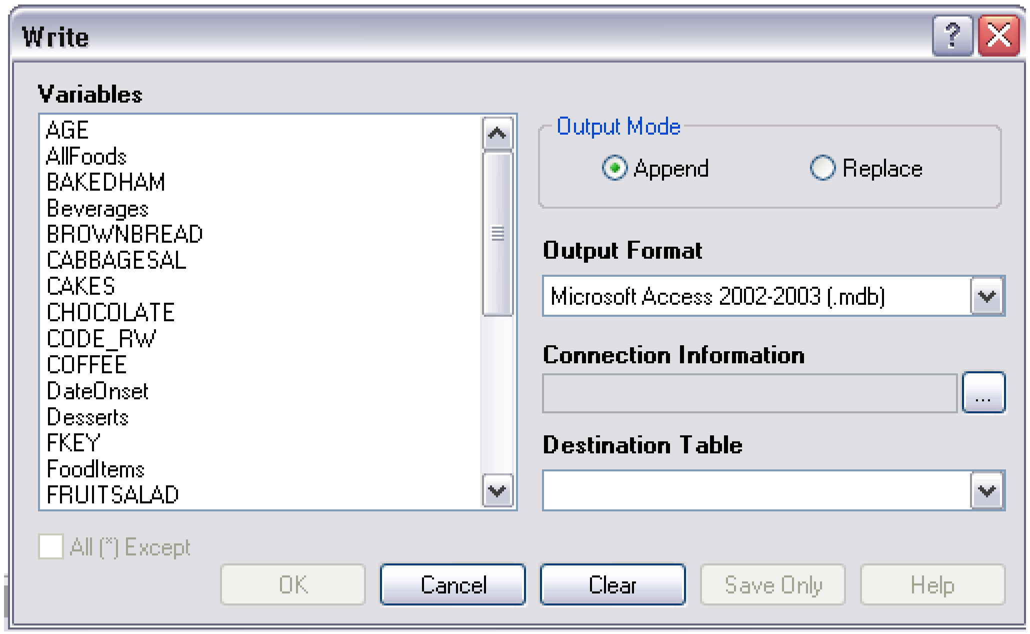


Figure 7.92: Write Dialog Box

2. Keep the default setting **All (*)** unchecked to write all the variables to a new file.

   Note: You can also choose individual variables from the file list by clicking on the variable name or select **All (*) Except** to exclude specific variables.

3. From the Output Mode section, click the **Replace** radio button.

- **Replace** creates a new file and overwrites any existing variables and records. To overwrite or replace data in a table, use the Replace selection.
- **Append** adds new variables and records to the end of existing data.

4. Using the drop-down list, select the desired **Output Format**.
    - Available output formats are the same as ones that can be imported using the READ command (i.e., MS Access, MS Excel, SQL Server and Flat ASCII file).
5. Click the **Browse** button in the Connection Information field to specify file name and location.
6. Click **OK**.
7. Establish a **name** for the Destination Table or select a **table name** from the drop down list (if the table already exists).
8. Click **OK**.
    - To see the data, use the READ command to open the newly-created project.
    - The WRITE command will not create a form and cannot create a data table to work with a form. To preserve forms, use the MERGE command.
    - Not all output formats support all possible input formats.

Use the MERGE Command

Use the MERGE command to join records. MERGE is only supported if the READ data source is an Epi Info 7 project. A merge requires a key, called the GLOBAL RECORD ID, which represents an internal matching variable inside both sets of data.

Syntax

```
MERGE <table specification>  <key(s)> <type>
```

1. From the Classic Analysis Command Tree, use the READ command to open **a PRJ project file**.
2. From the Classic Analysis Command Tree, click **Data > Merge**. The MERGE dialog box opens.

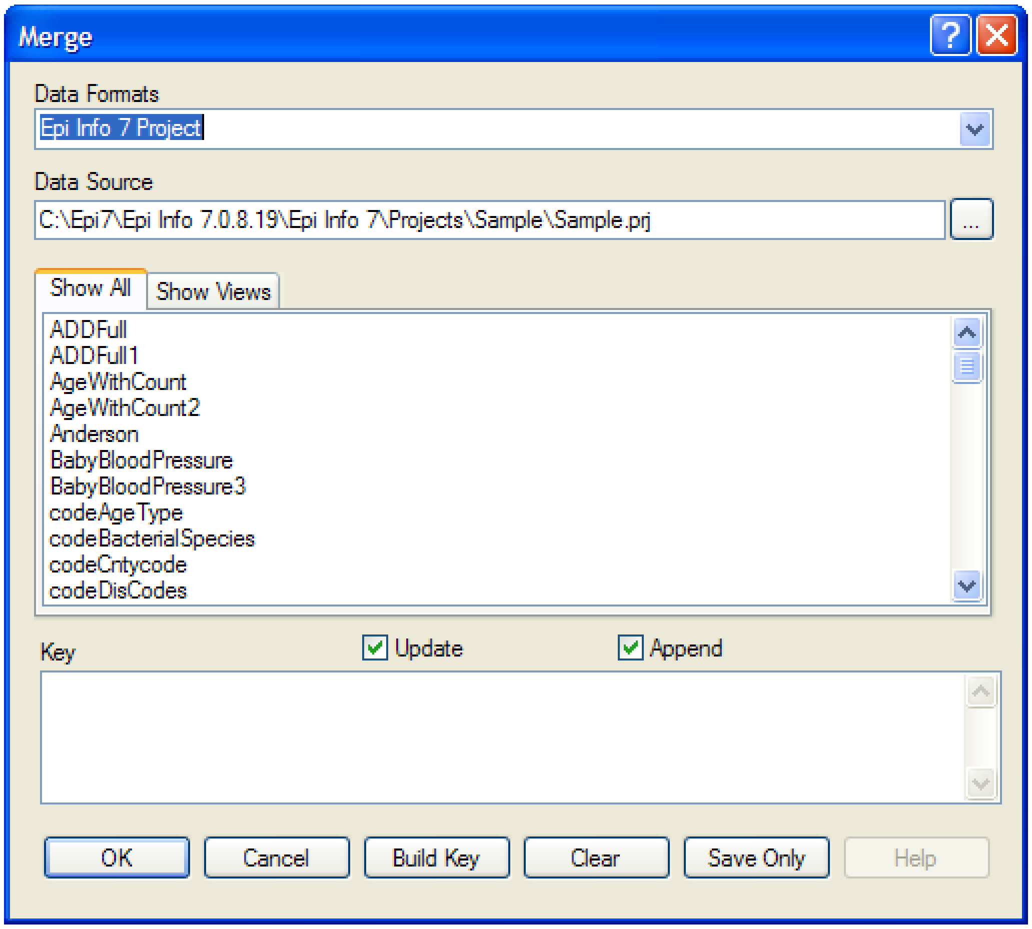


Figure 7.93: Merge Dialog Box

3. From the Data Formats drop-down list, specify the other Epi Info 7 project to be read. (Other data sources will be supported in a future release).
4. In the Data Source field, click the **Browse** button. The Merge File Name dialog box opens.
5. Select the **project file** that contains the data to be merged.
6. Click **Open**. The MERGE dialog box populates with available tables or worksheets.
7. Merge a **data table** or **worksheet**. The Build Key button activates.
8. Click **Build Key**. The RELATE - BUILD KEY dialog box opens.
    - There must be a variable in the Current Table that corresponds to a one in the Related Table (i.e., Patient ID).

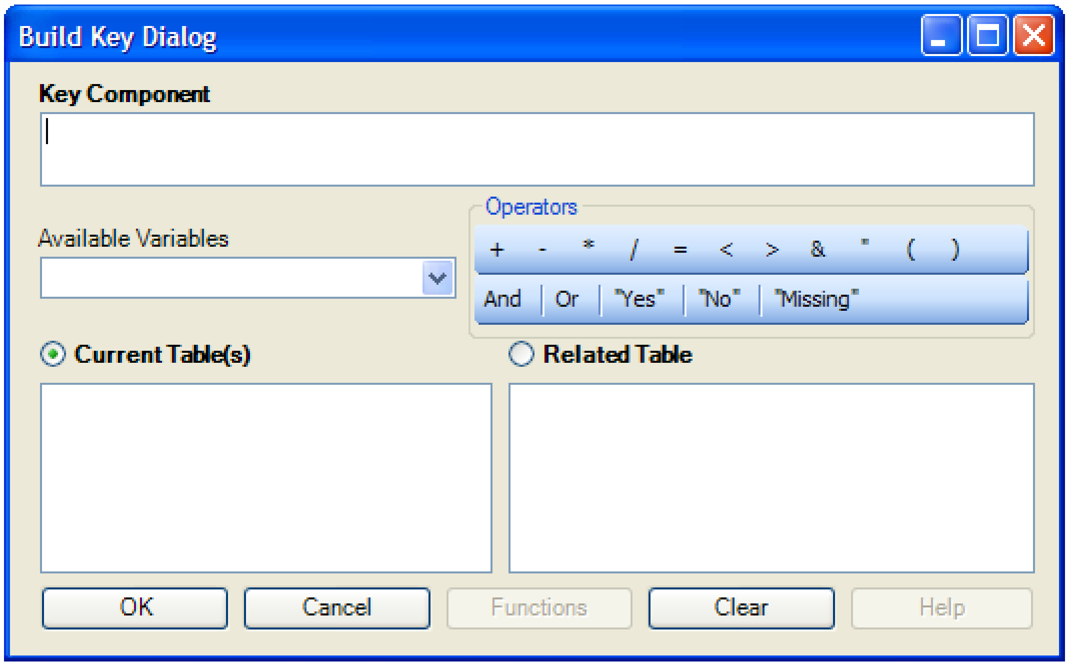


Figure 7.94: Build Key Dialog Box

9. From the Available Variables drop-down list, select the **current table merge variable.**

10. Click **OK**. The Current Table field shows the selection.

11. From the Available Variables drop-down list, select the **related table merge variable.**

12. Click **OK**. The Related Tables field shows the selection.

13. Click **OK** to accept the Build Key designations. The MERGE dialog box opens.

14. Make **Update** and **Append** selections from the corresponding checkboxes.

    - For matching records**,** Update replaces the value of any field in the READ table whose build key matches the MERGE table.
    - For unmatched records, Append creates a new record in the READ table with values only for those fields that exist in the MERGE table.
    - For most merges, select **Update** and **Append** records. If you select both options, records containing the same selected merge variables will be updated (overwritten) if there is new information. All other records will be appended (added) to the end of the data table.

15. From the Merge dialog box, click **OK**.

    - If the MERGE table is the same project as the READ table, the Record Count in the Output window updates to reflect any new records added to the table.

Use the ASSIGN Command

This command assigns an expression result or the field value to a variable. Variables are usually created with the DEFINE command and assigned a value.

Syntax

```
ASSIGN <variable> = <expression>
```

1. From the Classic Analysis Command Tree, click **Variables > Assign**. The ASSIGN dialog box opens.

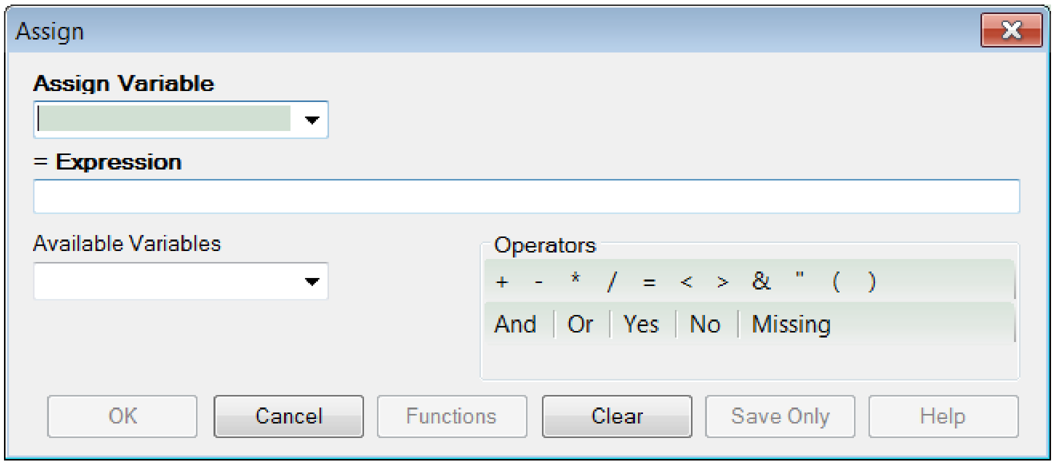


Figure 7.95: Assign Dialog Box

2. From the Assign Variable drop-down list, select the **variable** to have a value assigned.
3. In the =Expression field, create the **assignment syntax** based on needed data.
4. Click **OK**. The code appears in the Program Editor.

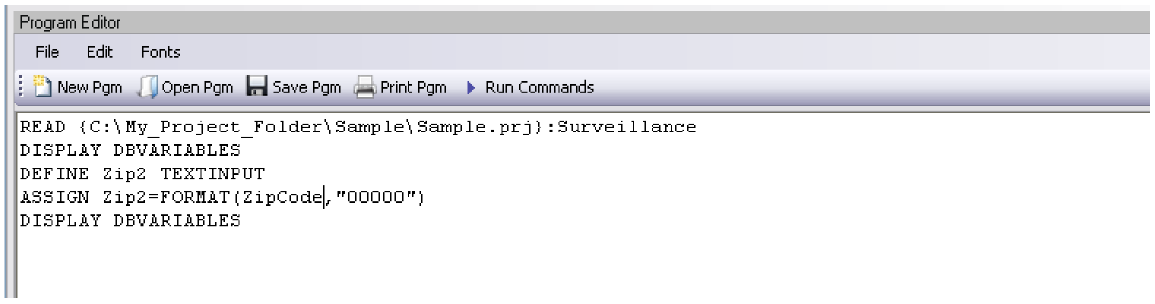
Try It

In the following example, the zip code field is a number. To use the zip code in a map, a new zip code variable must be defined and assigned text values.

11. From the Classic Analysis Command Tree, use the READ command to open the **Sample.PRJ project**.
12. From the Form section, click **Surveillance**.
13. Click **OK**.
14. Click **Variables > Display**.>--**Variables currently available**.

15. Click **OK**. The variables information appears.
16. Click **Variables > Define**. The DEFINE dialog box opens.
17. In the Variable Name field, create a new variable named **Zip2**.
18. Select **Text** for Variable Type
19. Click **OK**.
20. From the Classic Analysis Command Tree, click **Variables > Assign**. The ASSIGN dialog box opens.
21. From the Assign Variable drop-down list, select **Zip2**.
22. In the =Expression field, type the syntax **FORMAT(ZipCode,"00000")**.
    - In this example, the FORMAT function converts the format of the values of the variable ZipCode into text format. Text values are always surrounded by double quotes and assigned to the new Zip2 variable.

10. Click **OK**. The code appears in the Program Editor.
11. Use the DISPLAY command to view variable information.



```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Surveillance
DISPLAY DBVARIABLES
DEFINE Zip2 TEXTINPUT
ASSIGN Zip2=FORMAT(ZipCode,"00000")
DISPLAY DBVARIABLES
```

Figure 7.96: Variable Information in Program Editor

Delete File/Table

Delete Files

The file(s) specified explicitly or implicitly (via wildcards) are deleted. If no files are specified, or if any specified files cannot be deleted, a message is produced unless you select Run Silent. Wildcards are not allowed in the suffix.

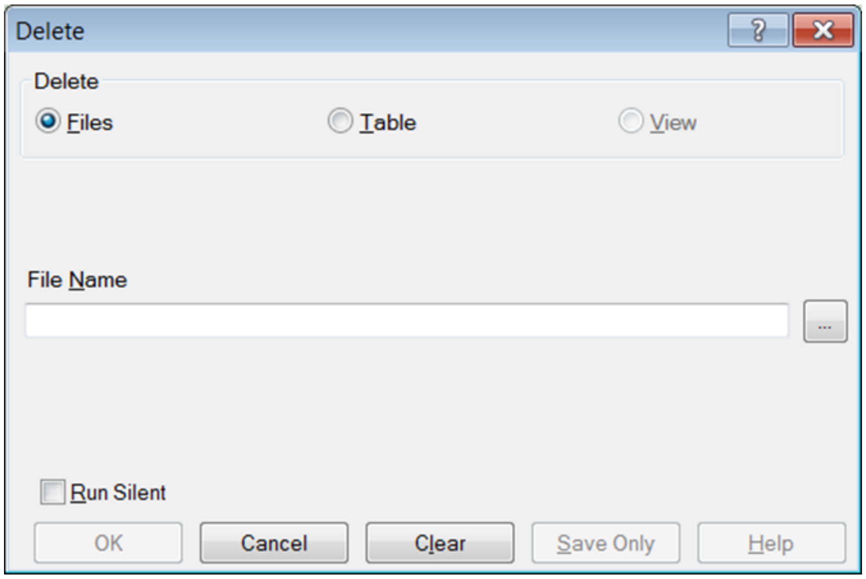


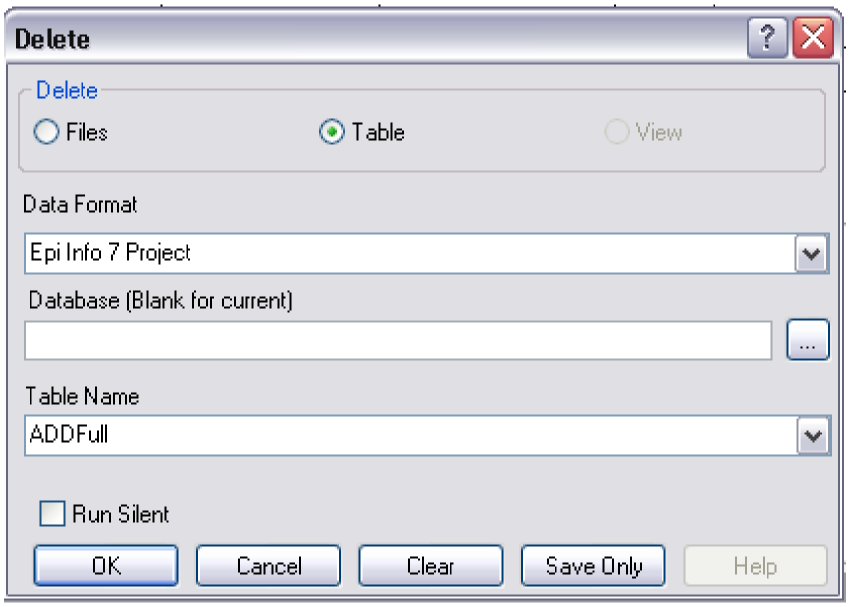
Figure 7.97: Delete Dialog Box

- **File Name** contains the name of the file to be deleted.
- **Run Silent** causes the command to run without any warning or error messages from the Program Editor.
- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Save Only** saves the created code to the Program Editor, but does not run the code.
- **Cancel** closes the dialog box without saving or executing a command.
- **Clear** empties the fields to allow information to be re-entered.
- **Help** opens the Help topic associated with the module being used (Currently Disabled).

Delete Table

The specified table is deleted. If the table does not exist or cannot be deleted, a message is produced unless you select Run Silent. To delete tables with spaces in their names, specify the file and the table even if the table is in the current project. The table must be enclosed in single quotes. It is possible to read a table with a space in its name and then delete it, resulting in errors during subsequent procedures.

DELETE TABLES will not delete and produce messages if any of the tables specified are data, grid, or program. Code tables are deleted only if they are not referenced by any form.



Delete: Files [✓ Table] View

Data Format: Epi Info 7 Project

Database (Blank for current):

Table Name: ADDFull

Run Silent

Figure 7.98: Delete Tables Dialog Box

- **Data Format** specifies the data source format of the file containing the table to be deleted.
- **Database (Blank or Current)** specifies the name and location of the dataset containing the table to be deleted.
- **Table Name** specifies the name of the table to be deleted.
- **Run Silent** causes the command to run without any warning or error messages from the Program Editor.
- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Save Only** saves the created code to the Program Editor, but does not run the code.

- **Cancel** closes the dialog box without saving or executing a command.
- **Clear** empties the fields to re-enter information.
- **Help** opens the Help topic associated with the module being used. (Currently Disabled).

Delete Records

Delete Records may not be used with related tables. For DELETE * and DELETE TABLES, space will not be reclaimed. You can run the Epi Info 3.5.3 Compact Database utility to reclaim space. All records in the current selection matching the expression are set to deleted status, or if PERMANENT is specified, physically deleted. Unless you select RUNSILENT, a confirmation message is displayed.

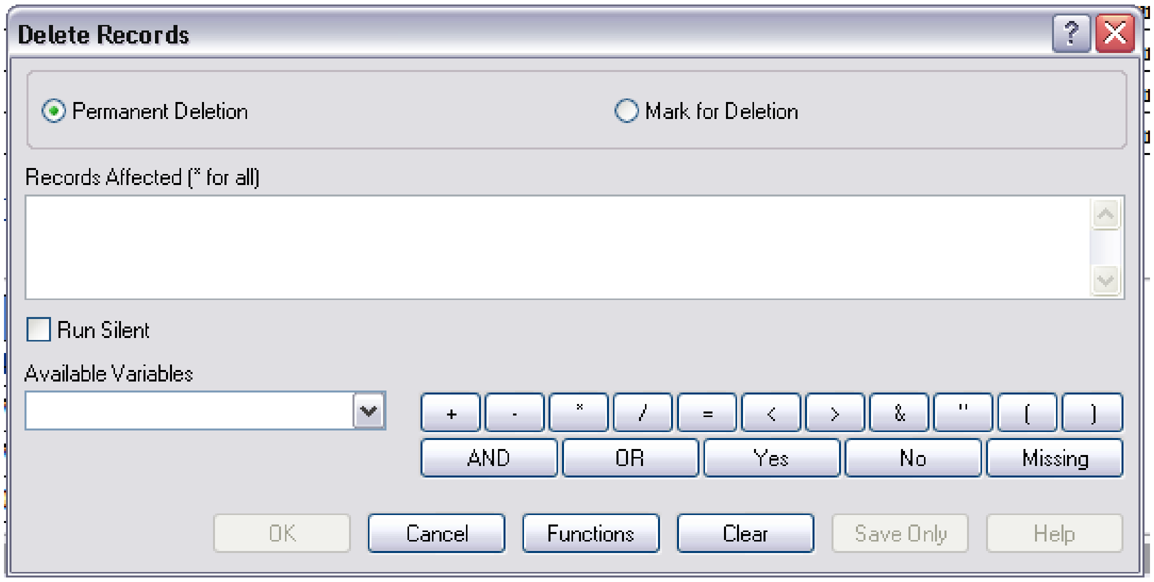


Figure 7.99: Delete Records Dialog Box

- **Permanent Deletion** causes records to be inaccessible after deletion.
- **Mark for Deletion** marks the selected records as to be deleted, and permits you to undelete. This feature can only be applied to Epi Info 7 projects.
- **The Records Affected (* for all)** field specifies which records to delete.
- **Run Silent** causes the command to run without any warning or error messages from the Program Editor.
- The **Available Variables** drop down allows you to select variables available in the current project.

- Functions and operators appear within commands and are used for common tasks (e.g., extracting a year from a date, combining two numeric values, or testing logical conditions).
- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Save Only** saves the created code to the Program Editor, but does not run the code.
- **Cancel** closes the dialog box without saving or executing a command.
- **Clear** empties the fields to re-enter information.
- **Help** opens the Help topic associated with the module being used. (Currently Disabled).

Use the ROUTEOUT Command

You can locate output by using the ROUTEOUT command. If no directory exists, the file is placed in the current project's directory. Results accumulate until you execute a CLOSEOUT command. Output files can be placed in any folder. The ROUTEOUT command selects a path and filename. If no output file is selected, Classic Analysis uses the default value. In each folder, Classic Analysis creates a new index table that contains links to the files created.

1. From the Classic Analysis Command Tree, use the READ command to open a **PRJ project file.**
2. From the Classic Analysis Command Tree, click **Output > RouteOut**. The ROUTEOUT dialog box opens.

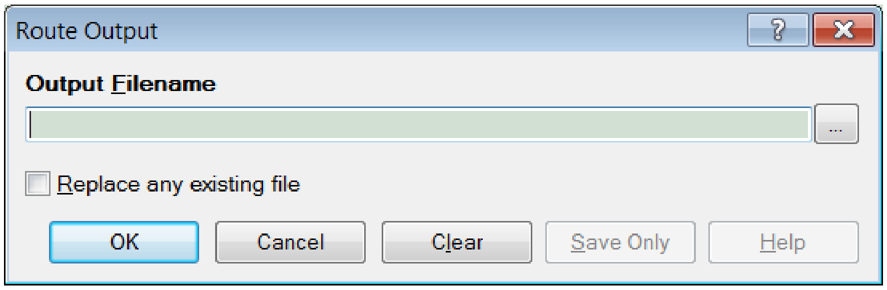


Figure 7.100: Route Output Dialog Box

3. In the Output Filename field, enter a **file name** to locate an existing file.
   - If necessary, select **Replace Existing File** to write over any files with the same name.
4. Click **OK**. Create **Output** on the existing project.
   - Notice the title bar contains the output filename specified in the ROUTEOUT command.
   - The new file is placed in the selected directory with the extension .HTM.
5. From the Output window toolbar, click **Open**. The Browse dialog box opens.
6. Locate and select the file created with ROUTEOUT. Click **Open**. The Output appears in the Output window.
   - The saved Output file can be opened by any application that can read an HTML file.

To end the ROUTEOUT command, choose one of the following options.

- From the Output folder, click **Closeout**.
- READ in a **new project**.
- Close **Epi Info 7**.

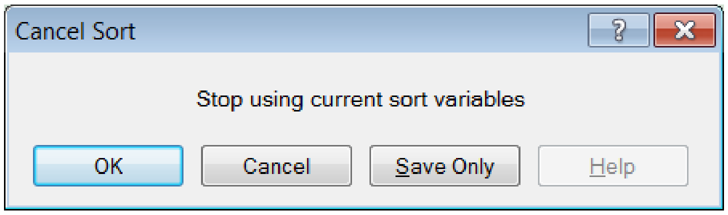
Use the CANCEL SORT Command

Syntax

CANCEL SORT Command Generator

CANCEL SORT Command Reference

1. From the Classic Analysis Command Tree, click **Select/If > Cancel Sort**. The CANCEL SORT dialog box opens.



Stop using current sort variables

Figure 7.101: Cancel Sort Dialog Box

2. Click **OK**. The selection criteria are removed from the data, and the original record count is restored.

Undelete Records

The UNDELETE command causes all logically deleted records in the current selection matching the expression to be set to normal status. This applies only to Epi Info 7 forms and may not be used when using related tables.

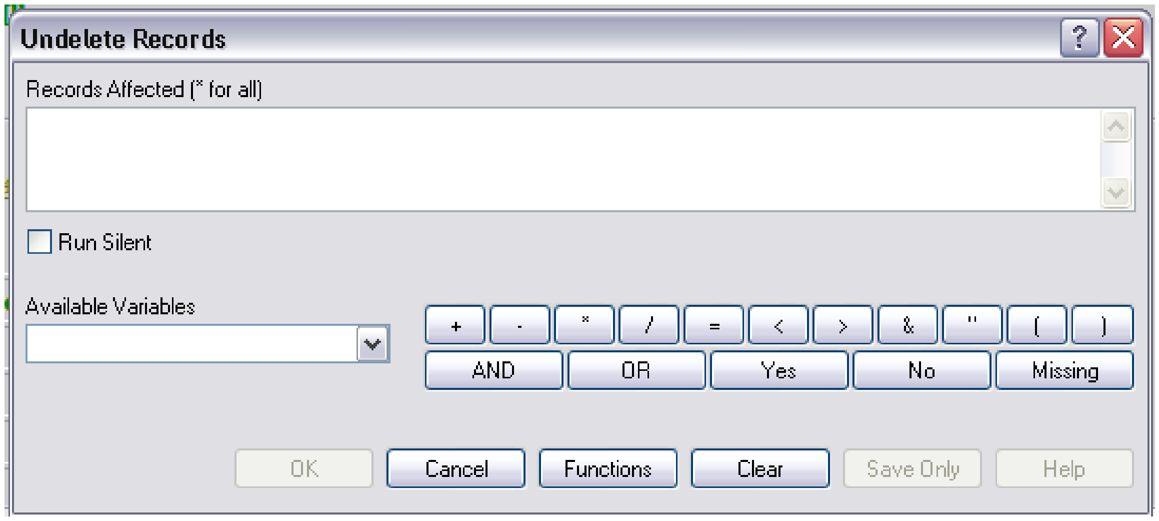


Figure 7.102: Undelete Records Dialog Box

- The **Records Affected (*for all)** field contains the expression for which records to undelete.
- The **Available Variables** field allows you to select available variables in the current project.
- Functions and operators appear within commands and are used for common tasks (e.g., extracting a year from a date, combining two numeric values, or testing logical conditions).
- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Save Only** saves the created code to the Program Editor, but does not run the code.
- **Functions** opens the online help file, which explains how to use functions and operators.
- **Cancel** closes the dialog box without saving or executing a command.
- **Clear** empties the fields to re-enter information.
- **Help** opens the Help topic associated with the module being used. (Currently Disabled).

How to Manage Variables

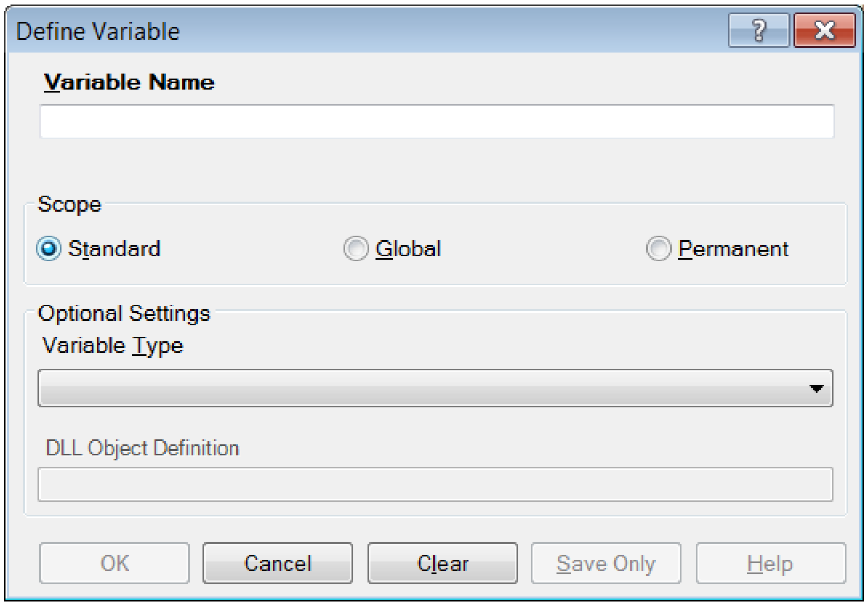
Use the DEFINE Command

This command creates new variables.

Syntax

```
DEFINE <variable> {<scope>} {<type indicator>} {("<prompt>")}
```

1. From the Classic Analysis Command Tree, click **Variables > Define**. The DEFINE Variable dialog box opens.



Scope: [✓ Standard] Global Permanent

Figure 7.103: Define Variable Dialog Box

2. From the Scope section, select one of the following variables:

   - **Standard** variables remain in a current project and are one variable in that form. They can assume new values with each record during a file query (e.g., a LIST or FREQ).
   - **Global** variables persist throughout program execution. They pass values from one form to a related form, and do not change during a file query.

- **Permanent** variables are stored in the EpiInfo.Config.xml file or system registry and retain any value assigned until changed by another assignment, or until the variable is undefined. They are shared among Epi Info 7 programs and persist even if the computer is shut down and restarted.

3. From the Variable Type drop-down list, select one of the following types: **Date**, **Numeric**, **Text**, or **Yes-No**.
4. Optional: Variable Type is required and cannot be used if <scope> is omitted. <type indicator> is the data type of the new variable and must be one of the following reserved words: NUMERIC, TEXTINPUT, YN, or DATEFORMAT. If omitted, the variable type will be inferred based on the data type of the first value assigned to the variable. However, omitting field type is not recommended.
5. Click **OK**.

Define a Group

To define a group, take the following steps:

1. From the Classic Analysis Command Tree, click **Variables > Define Group**. The DEFINE GROUP dialog box opens.
2. In the Group Variable field, type a **name**.
3. From the Variables list box, select the **variables** to be included in the group.
4. Click **OK**.

Convert a Number to Text

Numbers can be converted to text using the FORMAT function. FORMAT changes the format of one variable type to another.

Syntax

```
FORMAT(<variable>, {"<format specifcation>"})
```

Example

The Format function can be used to convert a numeric postal code or zip code field to a text type variable for mapping.

- The READ command opens a new file that contains a numeric field called zipcode. To use Epi Map with the zipcode values, change the values to text.
- The DEFINE command creates a new variable called NewZip. The variable NewZip will hold the values of zipcode in a text format. The code appears in the Program Editor as DEFINE NewZip.

- The ASSIGN command formats the variable NewZip with the values of ZipCode and the format of text. Text values are enclosed in quotes. The code appears in the Program Editor as:

```
ASSIGN NewZip=FORMAT(ZipCode,"00000")
```

- The variable NewZip can now be used as a geographic variable in Epi Map because it is a text type variable that contains numeric data.

Try It

For this example, use the Classic Analysis Program Editor to create the code.

1. Read in the **Sample.PRJ** project. Open **Oswego**. Seventy five (75) records should be listed in the Output window.
2. In the Program Editor window, place the cursor under the created Read command. Type **DEFINE MyAge TEXTINPUT**
    - Do not press the Enter key. Leave your cursor on the line of code you just created.
3. In the Program Editor window, type the following: **`ASSIGN MyAge = FORMAT(Age, "00").`**
4. From the Program Editor navigation menu, click **Run Commands**. The code will turn gray to indicate a syntax review.
5. From the Classic Analysis Command Tree, click **Statistics > List**. Click **OK**. The records appear.
    - Notice that the values in 'MyAge' are all two characters with a leading '0' before each single digit year. The leading zero is present because of the '00' format specification.

7. From the Classic Analysis Command Tree, click **Variables > Display**. Click **OK**. The variable types appear.
    - The variable MyAge now contains the values of Age and the format type of Text.

Use IF Statements with Permanent or Global Variables

- If the condition of an IF statement depends on global or permanent variables, the value is computed immediately and only the true or false branch, as appropriate, is followed.
- If the condition of an IF statement depends on a field variable, neither branch is followed. However, computations within the branches are saved for execution as appropriate on each record. In the second case, non-computational commands (e.g., READ, SELECT, SORT, HEADER, and ROUTEOUT) are never executed.

Handle Date Fields

Date formats are often determined by default computer settings. To access Regional Settings in a Windows environment, click **Start > Control Panel > Regional** and **Language Options**.

- Date fields in forms are entered in European, American, or ISO format as specified in Form Designer. They are stored in the database in a neutral format.
- Date fields in Excel, and Text files are read into Classic Analysis according to the date format found in Regional Settings, and stored in a neutral format. Excel files prepared with one date format often do not import properly on machines with a different date format.
- Date fields in forms are displayed in European, American, or ISO format as specified in Form Designer by the following Classic Analysis commands: LIST, LIST GRIDTABLE, TABLES, and FREQ.
- Date fields in forms are displayed according to the date format found in Regional Settings in the following Classic Analysis commands: DISPLAY, and GRAPH.
- Date fields in tables without forms are displayed according to the date format found in Regional Settings.
- Date fields written to Excel or Text files are written according to the date format in Regional Settings.
- Date literals in Form Designer, Check Code, or Classic Analysis Programs; and date-type permanent variables in the EpiInfo.Config.xml file, should be in American format unless enclosed in braces or pipe symbols. European date format (DD/MM/YYYY) literals must be enclosed in braces (i.e., {23/11/2007}). ISO date literals (YYYY/MM/DD) must be enclosed in pipe symbols (i.e.,|.,2007/11/23|.
- Dates formatted with the FORMAT function are formatted according to the date format found in Regional Settings when no format is specified as a second argument.
- Dates converted from text with TXTTODATE are converted to a neutral format according to the date format found in Regional Settings. For multi-national applications, it may be safer to use the NUMTODATE function.

If you are not sure, conduct experiments and examine the results.

How to Use Undefine

The Undefine command in Classic Analysis removes a defined variable from the system.

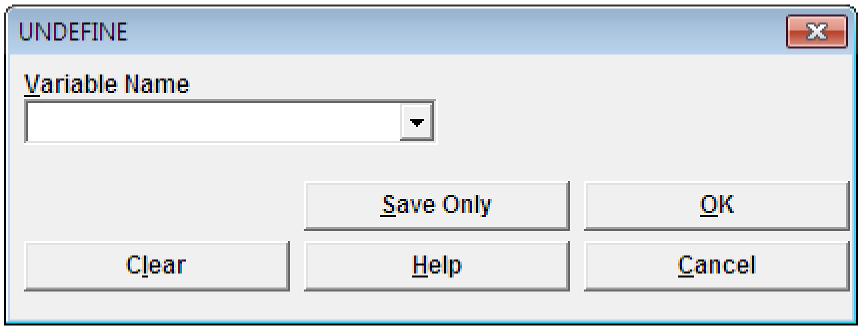


Figure 7.104: Undefine Command Box

- **Variable Name** indicates the name of the variable to be removed from the data table.
- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Save Only** saves the created code to the Program Editor, but does not run the code.
- **Cancel** command generation window without saving or executing a command.
- **Clear** empties fields so that information can be re-entered.
- **Help** opens the Help topic associated with the module being used. (Currently Disabled).

Use the ASSIGN Command

This command assigns the result of an expression or the field value to a variable. Variables are usually created with the DEFINE command and subsequently assigned a value.

Syntax

```
ASSIGN <variable> = <expression>

<variable> = <expression>
```

1. From the Classic Analysis Command Tree, click **Variables > Assign**. The ASSIGN dialog box opens.

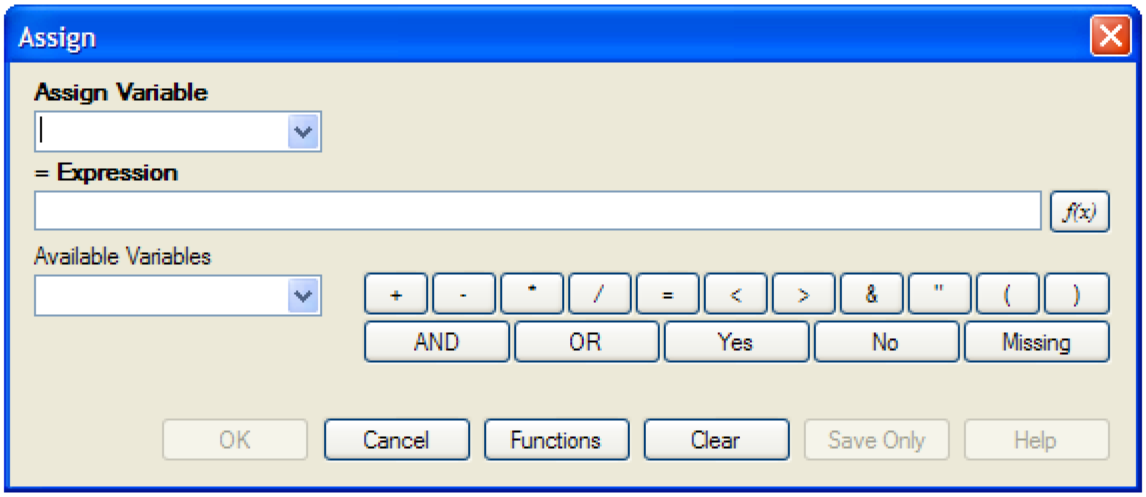


Figure 7.105: Assign Variable Dialog Box

10. From the Assign Variable drop-down list, select the **variable** to have a value assigned.

11. In the =Expression field, create the **assignment syntax** based on the needed data.

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

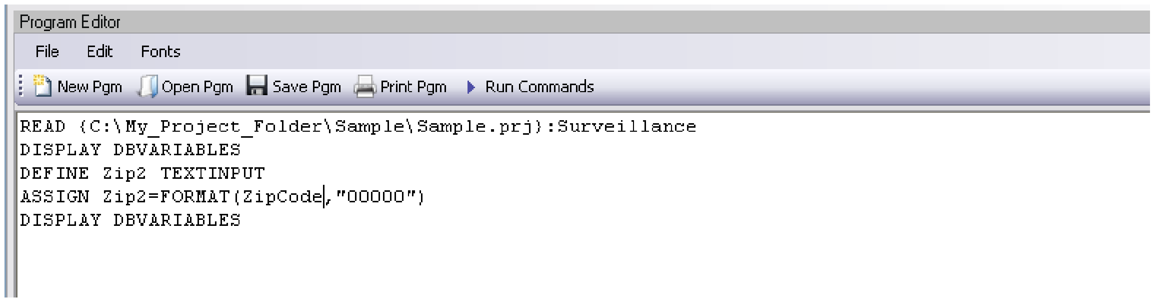
Try It

In the following example, the zip code field is a number. To use the zip code in a map, a new zip code variable must be defined and assigned text values.

1. From the Classic Analysis Command Tree, use the READ command to open the **Sample.PRJ project**.

2. From the form section, click **Surveillance**. Click **OK**.

3. Click **Variables > Display**. Click **OK**. The variables information appears.
4. Click **Variables > Define**. The DEFINE dialog box opens.
5. In the Variable Name field, create a new variable named **Zip2**.
6. Click **OK**.
7. From the Classic Analysis Command Tree, click **Variables > Assign**. The ASSIGN dialog box opens.
8. From the Assign Variable drop-down list, select **Zip2**.
9. In the =Expression field, type the syntax **`FORMAT(ZipCode,"00000")`.**
    - In this example, the FORMAT function is used to convert the format of the values of the variable ZipCode into the text format. Text values are always surrounded by double quotes and are assigned to the new Zip2 variable.
10. Click **OK**. The code appears in the Program Editor.
11. Use the DISPLAY command to view variable information.



```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Surveillance
DISPLAY DBVARIABLES
DEFINE Zip2 TEXTINPUT
ASSIGN Zip2=FORMAT(ZipCode,"00000")
DISPLAY DBVARIABLES
```

Figure 7.106: Assign Variables in Program Editor

How to Use Recode

The Recode command allows grouping of data for age and other variables, or changing code from one system to another.

- To insert a line in the table, select the **row** and press **Ctrl-Insert**.
- To delete a line in the table, select the **row** and press **Ctrl-Delete**.
- Delete any **blank rows** prior to selecting **OK** or **Save Only**.

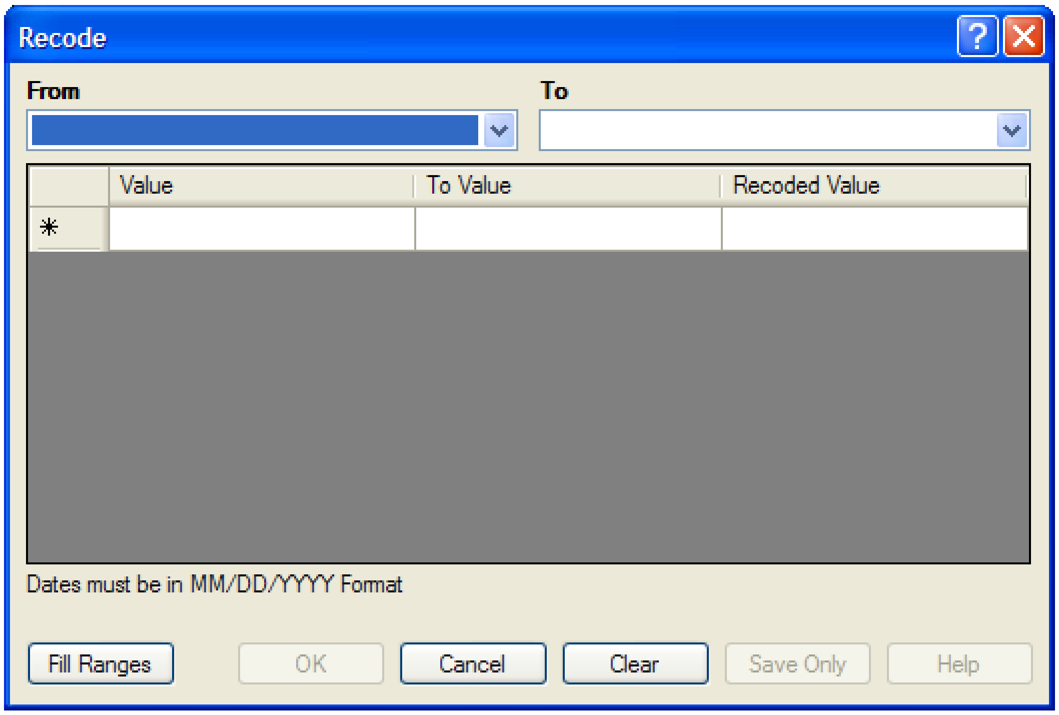


Figure 7.107: Recode Dialog Box

- The **From** drop down identifies the variable whose values are to be recoded.
- The **To** drop down identifies the variable slated to receive the recoded values.
- **Value (blank = other)** identifies the bottom of a range of values, a single value, or all remaining values slated to be recoded. Enclose text in quotation marks. The word LOVALUE may be used to indicate the smallest value in the database. HIVALUE may be used to indicate the largest. To recode a single value instead of a range, use only this column. To recode all unspecified values, leave this column and the To column blank.
- **To Value (if any)** identifies the top of a range of values that will be recoded.
- **Recoded Value** identifies the value to be assigned to the destination variable for the specified values of the source variable. Enclose text in quotation marks.

- **Fill Ranges** allows you to redefine recoded ranges of equal distribution.
- **OK** accepts the current settings and data, and closes the form or window.
- **Save Only** saves the created code to the Program Editor, but does not run the code.
- **Cancel** command generation window without saving or executing a command.
- **Clear** empties fields so information can be re-entered.
- **Help** opens the Help topic associated with the module being used. (Currently Disabled).

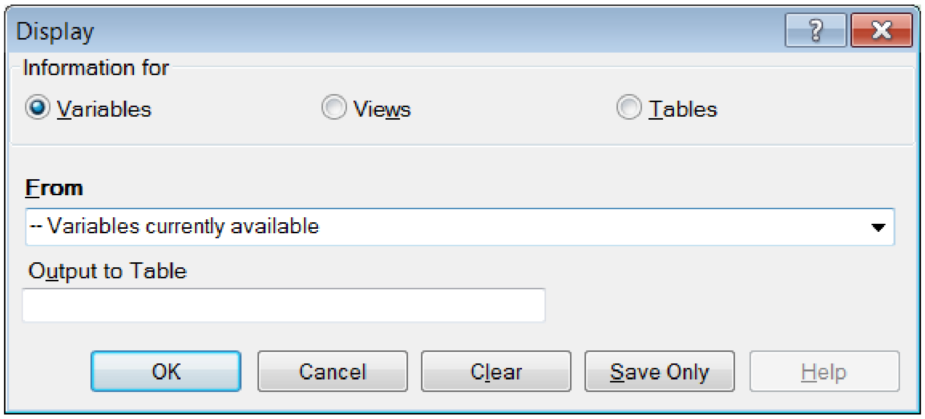
Use the DISPLAY Command

This command displays table, form, and database information.

Syntax

```
DISPLAY <option> [<sub-option>] [OUTTABLE=<tablename>]
```

1. From the Classic Analysis Command Tree, use the READ command to open a **PRJ project file**.
    - If a .PRJ file is not opened, only language is shown for selected variables.
2. From the Classic Analysis Command Tree, click **Variables > Display**. The DISPLAY dialog box opens.



Variables [✓] selected; From: -- Variables currently available; Output to Table: (empty)

Figure 7.108: Display Dialog Box

3. From the Information for section, select one of the following form data options:
    - **Variables** displays information about all of the variables in the dataset; the defined, field, and selected variables.
    - **Forms** displays the forms in the current dataset or a selected dataset. Information about forms and other Epi Info-specific tables is shown.
    - **Tables** displays information about tables in a database. Information about tables and other Epi Info-specific or generic tables is shown.
4. If needed, type an **Output to Table** name.
5. Click **OK**. The information is displayed in the Output window.

How to Select Records

Use the SELECT Command

Use SELECT to view or analyze specific records based on selection criteria.

Syntax

```
SELECT <expression>
```

1. From the Classic Analysis Command Tree, click **Select/If > Select**. The SELECT dialog box opens.

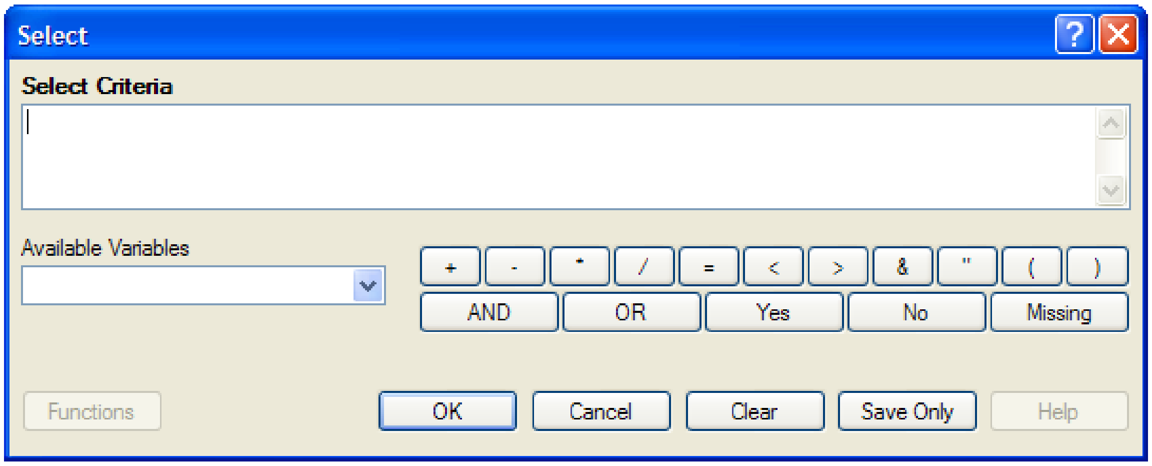
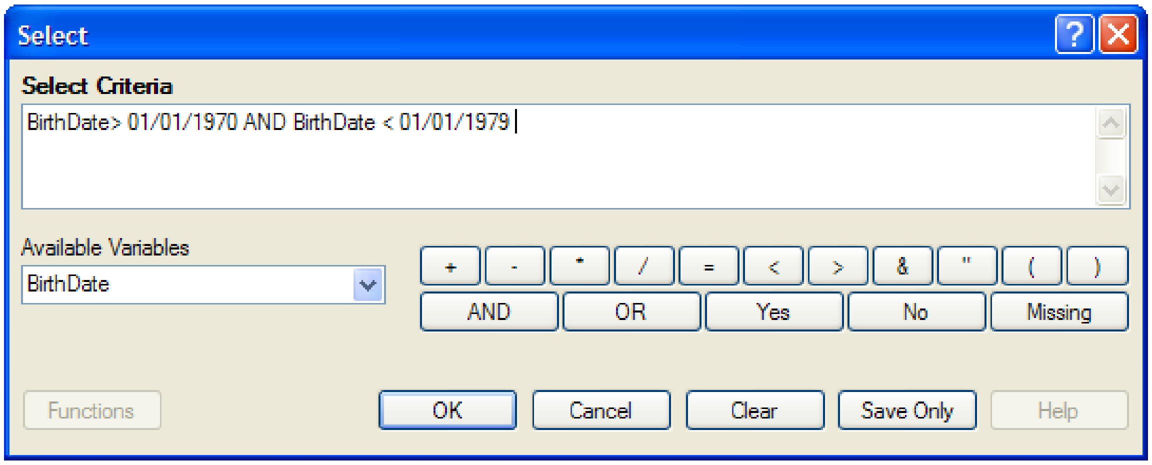


Figure 7.109: Blank Select Dialog Box

2. Use the Available Variables drop-down list to select variables from the open dataset.
3. Use the Select Criteria field and the Functions and Operators buttons to create selection code.
4. Click **OK**. The Record Count in the Output window should alter based on the number of records meeting the new criteria.
5. From the Classic Analysis Command Tree, click **Statistics > List** to view the records matching the selection criteria.
6. From the Classic Analysis Command Tree Select/If folder, click **Cancel Select** to cancel the selection criteria and return the Record Count to all records in the dataset.

Select a Date Range

1. From the Classic Analysis Command Tree, use the READ command to open a **PRJ project file**.
2. From the Classic Analysis Command Tree, click **Select/If > Select**. The SELECT dialog box opens.



BirthDate> 01/01/1970 AND BirthDate < 01/01/1979

BirthDate

Figure 7.110: Select Dialog Box

3. From the Available Variables drop-down list, select a **date variable**.
   - In this example, DateVar is a date variable used in the data table. The DateVar has to be a date field or be converted into a date field using the NUMTODATE or TXTTONUM functions.
4. In the Select Criteria field, create the date range selection code by using the greater than **>**, less than **<**, and AND operators.

Note: Unless the date literal is enclosed in braces or pipe symbols, literal dates in PGMs are always in U.S. date format. European date format (DD/MM/YYYY) literals must be enclosed in braces (i.e., {23/11/2007}). ISO date literals (YYYY/MM/DD) must be enclosed in pipe symbols (i.e.,|.,2007/11/23|).

Create the following example using the project called Sample.PRJ and the data from Surveillance.

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Surveillance
SELECT BirthDate > 01/01/1970 AND BirthDate < 01/01/1979
```

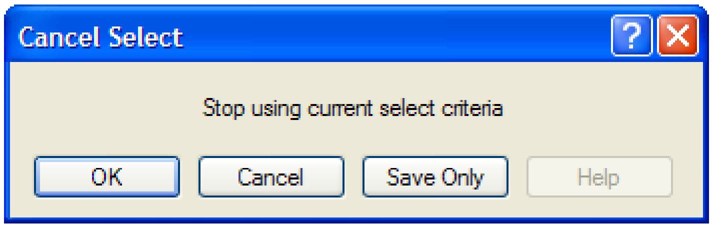
Use the CANCEL SELECT Command

SELECT

CANCEL SELECT Command Generator

CANCEL SELECT Command Reference

1. From the Classic Analysis Command Tree, click **Select/If > Cancel Select**. The CANCEL SELECT dialog box opens.



Stop using current select criteria

Figure 7.111: Cancel Select Dialog Box

2. Click **OK**. The selection criteria are removed from the data and the original record count is restored.

Use the IF Command

The IF command defines a condition. True causes the THEN code block to be executed; false causes the ELSE code block (if present) to be executed.

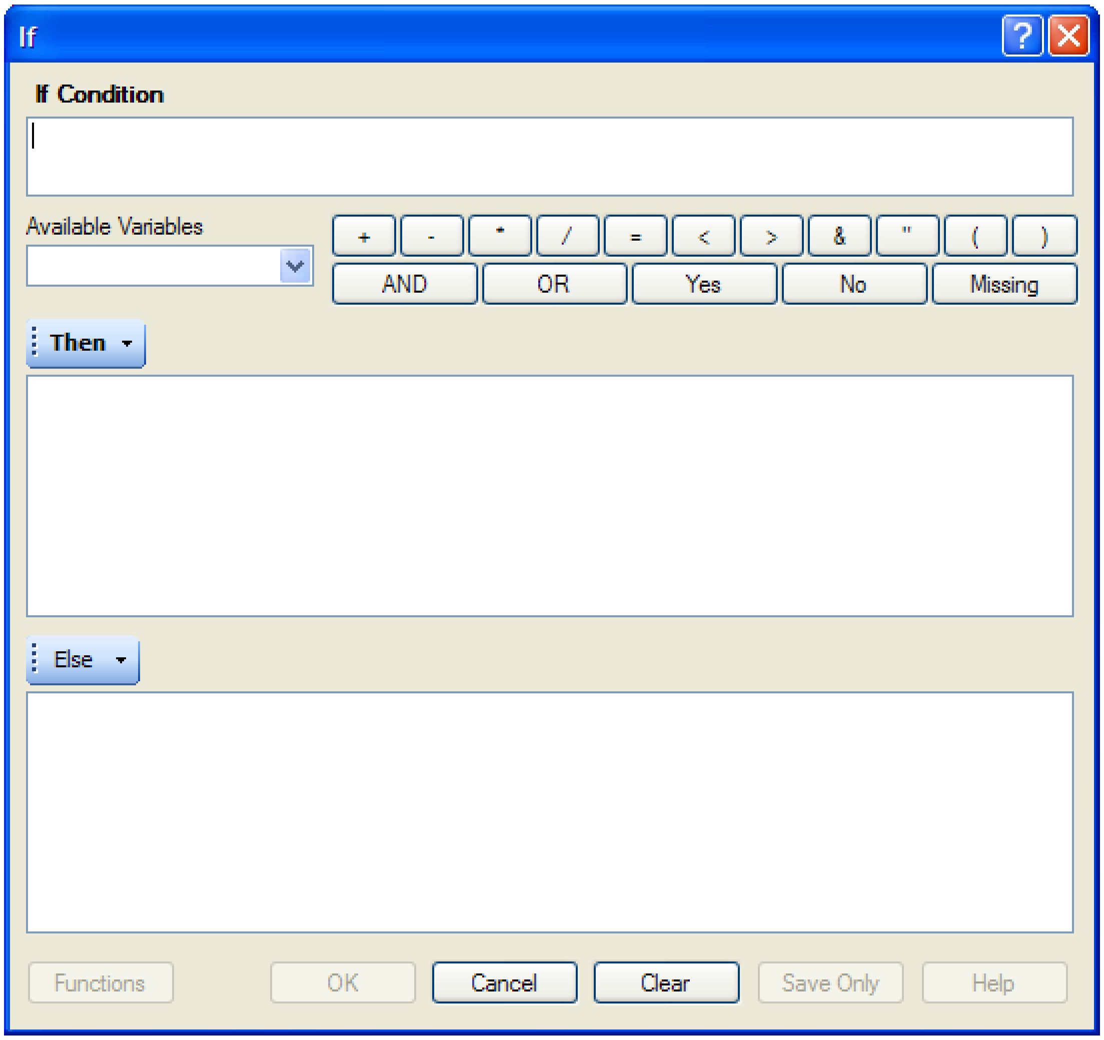


Figure 7.112: If Command Dialog Box

- The **If Condition** field specifies the condition to determine which statement that follows it is executed.
- The **Available Variables** field allows you to select variables available in the current project.
- The **Then** button is part of the If Condition statement. It starts the section of code executed when the If condition is true.

- The **Else** button is part of the If Condition and works as follows: If the condition is true, the first statement is executed. If false, the statement is bypassed, and if an else statement exists, it is executed instead.
    - Functions and operators appear within commands and are used for common tasks (e.g., extracting a year from a date, combining two numeric values, or testing logical conditions).
    - **OK** accepts the current settings and data, and subsequently closes the form or window.
    - **Save Only** saves the created code to the Program Editor, but does not run the code.
    - **Functions** opens the online Help file which explains the functions and operators. (Currently Disabled).
    - **Cancel** closes the dialog box without saving or executing a command.
    - **Clear** empties the fields so information can be re-entered.
    - **Help** opens the Help topic associated with the module being used. (Currently Disabled).

Use the Sort Command

The Sort command allows you to specify a sequence for records to appear when using the LIST, GRAPH, and WRITE commands.

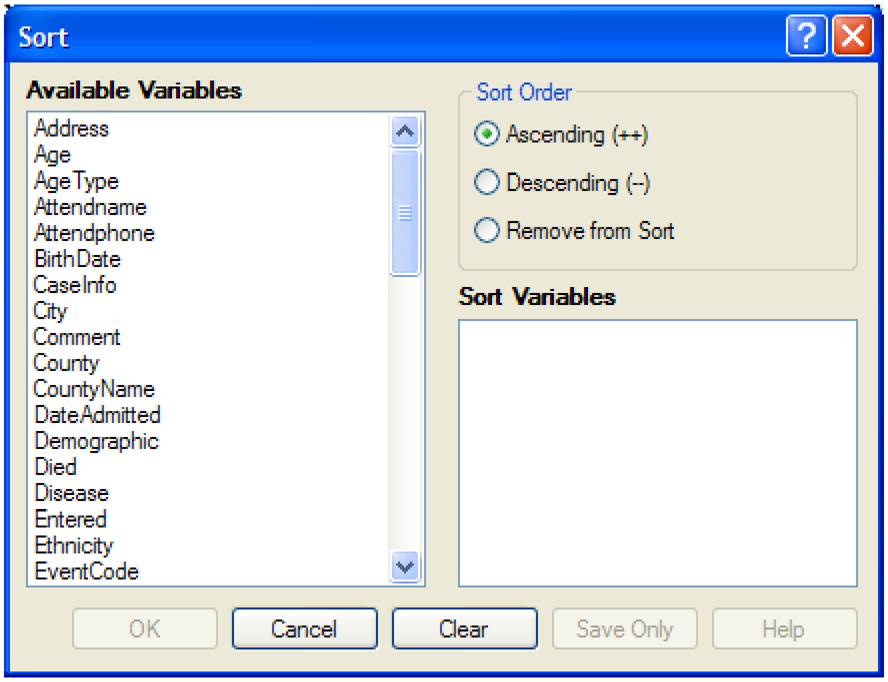


Figure 7.113: Sort Command Dialog Box

- The **Available Variables** field allows you to select variables available in the current project.
- **Sort Order** allows variables to be sorted in ascending (A to Z) or descending (Z to A) order. If the order is not specified, the sort order will be ascending. To sort one or more variables in descending order, the descending parameter (--) must be after each variable.
- Select **Remove from Sort** to remove the variables selected from the list.
- **Sort Variables** contains a list of variables selected for the sort.
- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Save Only** saves the created code to the Program Editor, but does not run the code.
- **Cancel** closes the dialog box without saving or executing a command.
- **Clear** empties the fields so that information can be re-entered.
- **Help** opens the Help topic associated with the module being used. (Currently Disabled).

CANCEL SORT

How to Use the CANCEL SORT command

The Cancel Sort command in Classic Analysis cancels a previous SORT command.

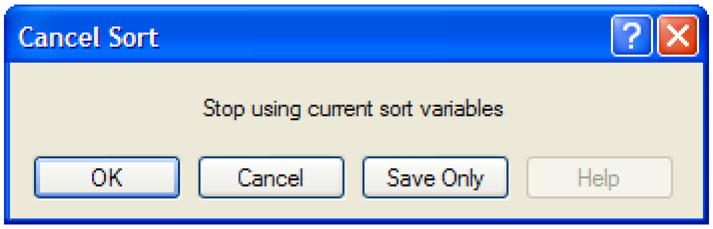


Figure 7.114: Cancel Sort

- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Save Only** saves the created code to the Program Editor, but does not run the code.
- **Cancel** closes the dialog box without saving or executing a command.
- **Help** opens the Help topic associated with the module being used. (Currently Disabled).

How to Display Statistics and Records

Use the LIST Command

The LIST command creates a line listing of the current dataset. Select specific variables from the Variables drop-down to narrow the list or select * (the Wild Card) to display all the variables. Check the **All (*) Except** box and select from the Variables drop-down to exclude variables from the list.

To navigate LIST * GRIDTABLE results, use the Tab key to move forward one cell at a time and the Shift+Tab keys to move back one cell at a time. Navigate through the records and cells using the left, right, up, and down arrow keys.

Syntax

```
LIST {* EXCEPT} [<variable(s)>] LIST {* EXCEPT} [<variable(s)>] {GRIDTABLE}
```

1. From the Classic Analysis Command Tree, use the READ command to open a **PRJ** project file.
2. From the Classic Analysis Command Tree, click **Statistics > List**. The LIST dialog box opens.

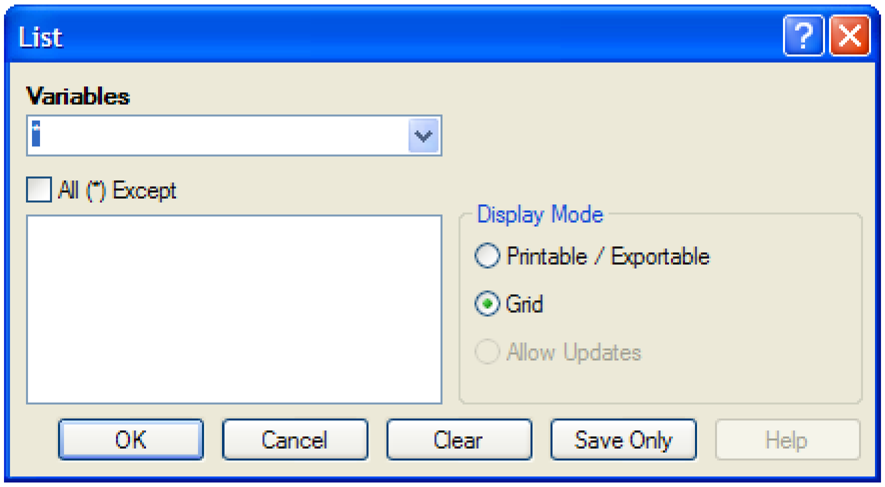


Figure 7.115: List Dialog Box

3. Click **OK** to accept the default settings. The variables in the dataset appear in a grid table inside the Output Window.
   - Grid output is not embedded. An Output file is not created.

Create a Web Format List

1. From the Classic Analysis Command Tree, click **Statistics > List**. The LIST dialog box opens.
2. From the Display Mode section, select the **Printable / Exportable** radio button.
3. Click **OK**. The List appears in the Output window in a web table format. Web display mode creates an embedded output file.

Use the FREQ Command

The FREQ command produces a frequency table that shows how many records have a value for each variable, the percentage of the total, and a cumulative percentage. The command FREQ * creates a table for each variable in the current form other than unique identifiers. This command is used to begin analyses on a new data set.

Syntax

```
FREQ [<variable(s)>]

FREQ * {EXCEPT [<variable(s)>]}
```

1. From the Classic Analysis Command Tree, use the READ command to open a **PRJ project file**.
2. From the Classic Analysis Command Tree, click **Statistics > Frequencies**. The FREQ dialog box opens.

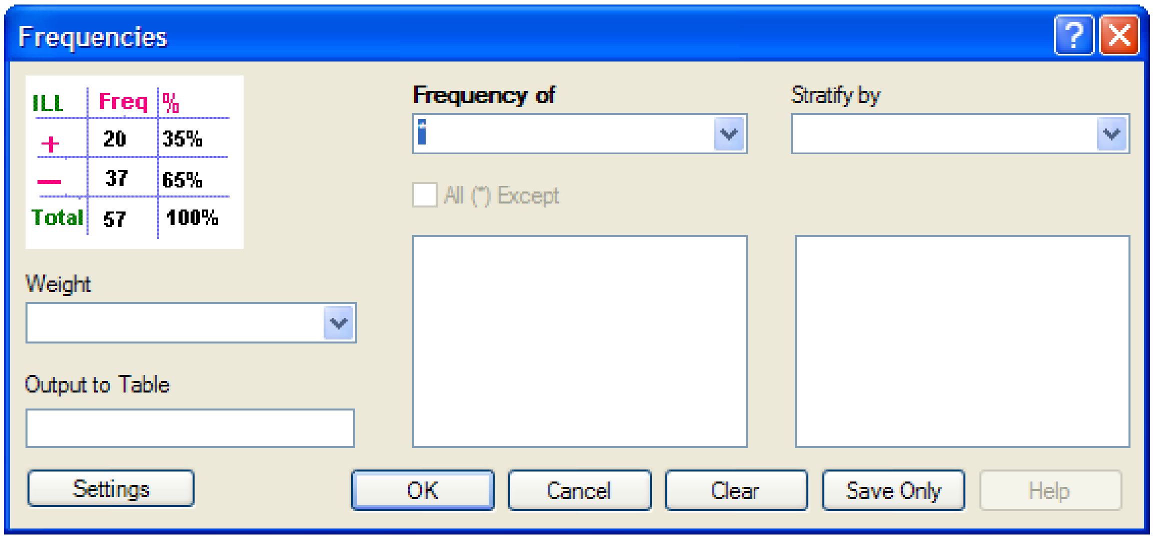


Figure 7.116: Frequency Dialog Box

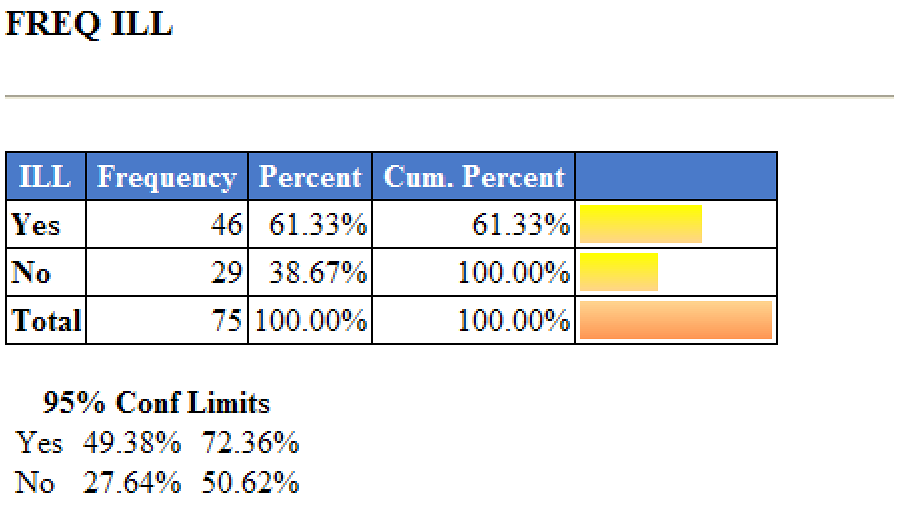
3. From the Frequency of drop-down list, select a **variable** from the data table.
   - Click the **All (*) Except** checkbox to exclude variables.
   - Use the corresponding drop-down lists to select a **Weight variable** or a **Stratify by** variable from the data source.
   - Use the **Output to Table** field to specify a location for table results. The new table can be accessed using the READ command.

4. Click **OK**. Results appear in the Output window.

Try It

For this example, use the Classic Analysis Command Tree to generate the FREQ command.

1. Read in the **Sample.PRJ** project. Open **Oswego**. Seventy five (75) records should be listed in the Output window.
2. Click **Frequencies**. The FREQ dialog opens.
3. From the Frequency of drop-down list, select **ILL**.
4. Click **OK**. Results appear in the Output window.



FREQ ILL

ILL	Frequency	Percent	Cum. Percent	
Yes	46	61.33%	61.33%	
No	29	38.67%	100.00%	
Total	75	100.00%	100.00%	

95% Conf Limits		
Yes	49.38%	72.36%
No	27.64%	50.62%

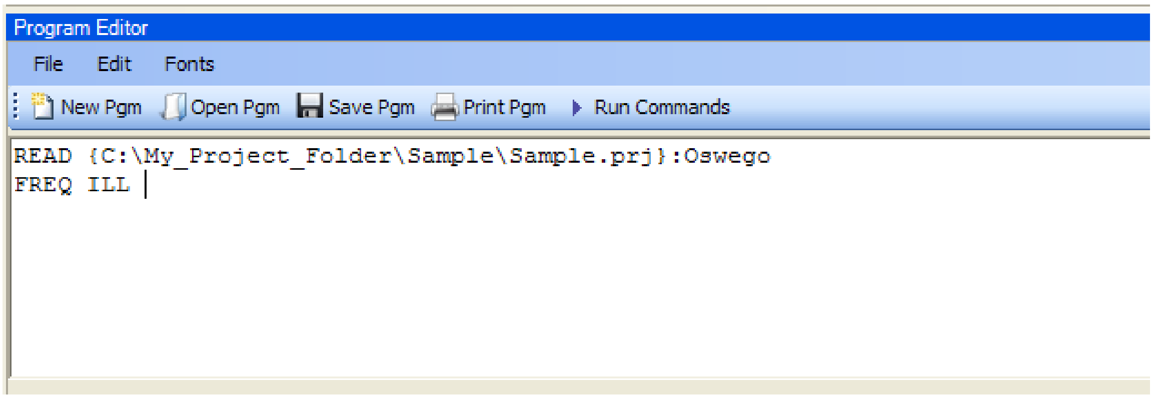
Figure 7.117: Frequency Output Window

- The **Frequency** column provides the count of individuals that were ill and not ill. The Percent column indicates the percentage of who were ill or not ill.
- The **95% Confidence Limits** are a range of values that indicates the likely location of the true value of a measure, meaning (in this instance) that the number of ill could be as low as 49.38%, or as high as 72.36%. Note that the Yes Ill percentage of 61.33% falls within the 95% Confidence Limits range of values based on the data.

Try It

For this example, use the Classic Analysis Program Editor to create the code.

1. Read in the **Sample.PRJ** project. Open **Oswego**. Seventy five (75) records should be listed in the Output window.
2. In the Program Editor window, place the cursor under the **Read command** created. Type **FREQ ILL**.
    - Do not press the Enter key. Leave the cursor on the line of code just created.



```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego
FREQ ILL
```

Figure 7.118: Frequency Command in Program Editor

3. Click **Run Commands** from the Program Editor navigation menu. The code turns green indicating syntax review and execution.
    - Results of the command appear in the Output window.
    - Results are the same as those created using the Command Tree dialog boxes.

Try It

A program or PGM can be saved and run repeatedly against a dataset for continuously updated statistics and analyses.

1. Read in the **Sample.PRJ** project. Open **Oswego**. Seventy five (75) records should be listed in the Output window.
2. Click **Open** from the Program Editor. The Read Program dialog box opens.
3. From the Program drop-down list, select **Statistics**.
4. Click **OK**. The PGM opens in the Program Editor.
5. Click **Run**. The Program Editor runs the code and output is placed in the Classic Analysis Output window.
    - Use the scroll bar to review all the results computed to the Output window.

Use the TABLES Command

The Tables command examines the relationship between two or more categorical values. For 2x2 tables, the Tables command produces odds and risk ratios. A 2x2 table is created when each selected variable has a yes or no answer.

Syntax

```
TABLES <exposure> <outcome> {STRATAVAR=[<variable(s)>]} {WEIGHTVAR=<variable>}
{PSUVAR=<variable>} {OUTTABLE=<table>} }
```

1. From the Classic Analysis Command Tree, use the READ command to open a **PRJ project file**.
2. From the Classic Analysis Command Tree, click **Statistics > Tables**. The TABLES dialog box opens.

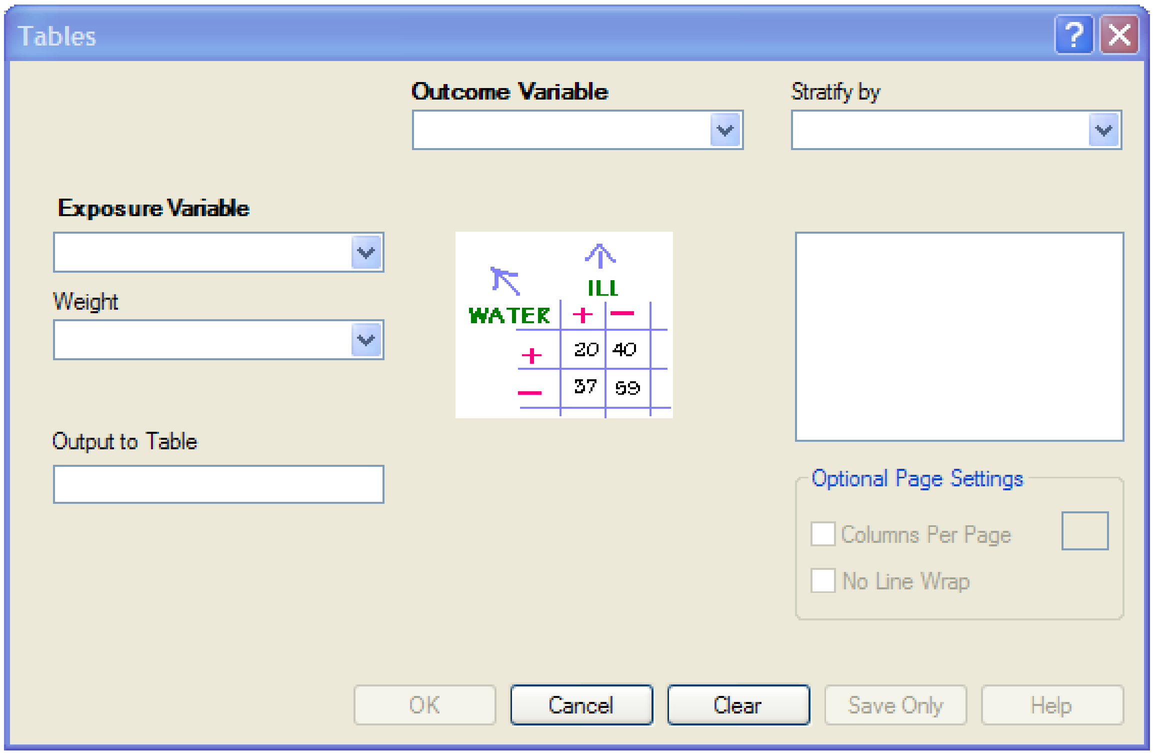


Figure 7.119: Tables Dialog Box

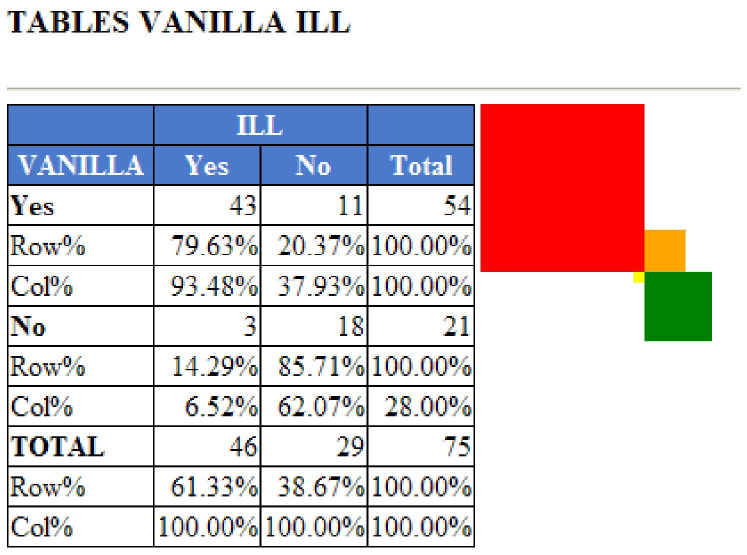
3. From the Exposure Variable drop-down list, select a **variable** from the data to indicate a risk factor exists.

4. From the Exposure Variable drop-down list, select a **variable** from the data to indicate the presence of an outcome. Use the:
   - Stratify by drop-down list to select a **variable** to act as a grouping variable.
   - Weight drop-down list to select a **variable** for weighted Classic Analysis.
   - Output to Table field to specify a location for table results.
   - READ command to access the READ command.
6. Click **OK**. Results appear in the Output window.
7. Scroll down to view the Single Table Classic Analysis.

Try It

Create a 2x2 table using data from the Sample.MDB project.

1. Read in the **Sample.PRJ** project. Open **Oswego**. Seventy five (75) records should be listed in the Output window.
2. Click **Tables**. The TABLES dialog box opens.
3. From the Exposure Variable drop-down list, select **Vanilla**.
4. From the Outcome Variable drop-down list, select **ILL**.
5. Click **OK**. Results appear in the Output window.

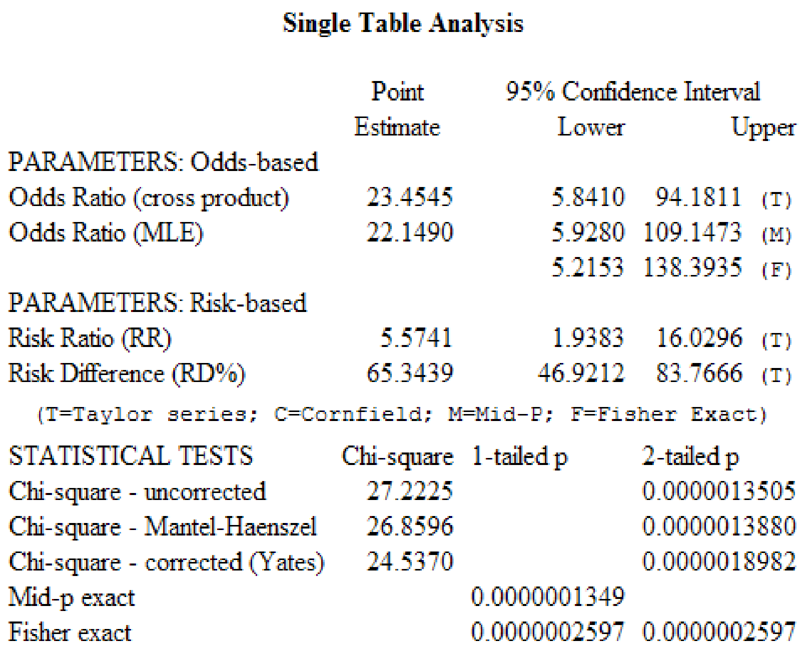


TABLES VANILLA ILL

	ILL		
VANILLA	**Yes**	**No**	**Total**
Yes	43	11	54
Row%	79.63%	20.37%	100.00%
Col%	93.48%	37.93%	100.00%
No	3	18	21
Row%	14.29%	85.71%	100.00%
Col%	6.52%	62.07%	28.00%
TOTAL	46	29	75
Row%	61.33%	38.67%	100.00%
Col%	100.00%	100.00%	100.00%

Figure 7.120: Tables Output Window

6. Scroll down to view the Single Table Classic Analysis.



Single Table Analysis

	Point Estimate	95% Confidence Interval Lower	Upper	
PARAMETERS: Odds-based				
Odds Ratio (cross product)	23.4545	5.8410	94.1811	(T)
Odds Ratio (MLE)	22.1490	5.9280	109.1473	(M)
		5.2153	138.3935	(F)
PARAMETERS: Risk-based				
Risk Ratio (RR)	5.5741	1.9383	16.0296	(T)
Risk Difference (RD%)	65.3439	46.9212	83.7666	(T)

(T=Taylor series; C=Cornfield; M=Mid-P; F=Fisher Exact)

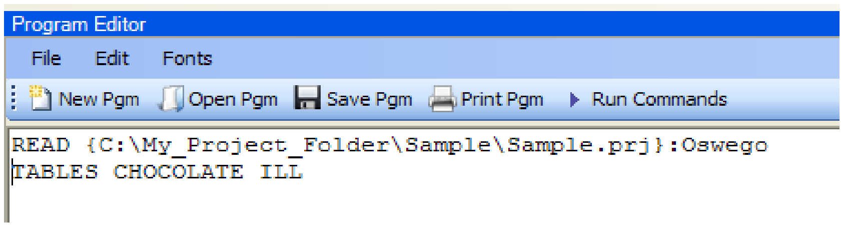
STATISTICAL TESTS	Chi-square	1-tailed p	2-tailed p
Chi-square - uncorrected	27.2225		0.0000013505
Chi-square - Mantel-Haenszel	26.8596		0.0000013880
Chi-square - corrected (Yates)	24.5370		0.0000018982
Mid-p exact		0.0000001349	
Fisher exact		0.0000002597	0.0000002597

Figure 7.121: Single Tables Classic Analysis

Try It

For this example, use the Classic Analysis Program Editor to create the code.

1. Read in the **Sample.PRJ** project. Open **Oswego**
2. In the Program Editor window, place the **cursor** under the Read command created. Type **TABLES Chocolate ill**.
   - Do not press the Enter key. Leave the cursor on the line of code just created.



```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego
TABLES CHOCOLATE ILL
```

Figure 7.122: Tables Command in Program Editor

3. Click **Run Commands** from the Program Editor navigation menu. The code turns gray indicating syntax review and execution.
    - Results of the command appear in the Output window.
    - Results are the same as those created using the Command Tree dialog boxes.

Try It

A program or PGM can be saved and run repeatedly against a dataset for continuously updated statistics and analyses.

1. To see an example of a PGM, read in the **Sample.PRJ** project. Open **Oswego**.
2. From the Program Editor, click **Open Pgm**. The Read Program dialog box opens.
3. From the Program drop-down list, select **Statistics**.
4. Click **OK**. The PGM opens in the Program Editor.
5. Click **Run Commands**. The Program Editor runs the code and output is placed in the Classic Analysis Output window. Use the scroll bar to review all the results computed to the Output window.

Use the MEANS Command

The MEANS command can obtain an average for a continuous numeric variable. Since Yes equals '1' and No equals '0', the mean of a yes-no variable is the proportion of respondents answering yes. For this situation, use the FREQ command. It has two formats. If only one variable is supplied, the program produces a table similar to one produced by FREQUENCIES with descriptive statistics. If two variables are supplied, the first is numeric containing data to be analyzed. The second indicates how groups will be distinguished. The output of this format is a table similar to one produced by TABLES with descriptive statistics of the numeric variable for each group variable value.

Syntax

```
MEANS <variable 1> {<variable 2>} {STRATAVAR=<variable(s)>} {WEIGHTVAR=<variable>}
{OUTTABLE=<tablename>}
```

4. From the Classic Analysis Command Tree, use the READ command to open a **PRJ project file**.
2. From the Classic Analysis Command Tree, click **Statistics > Means**. The MEANS dialog box opens.

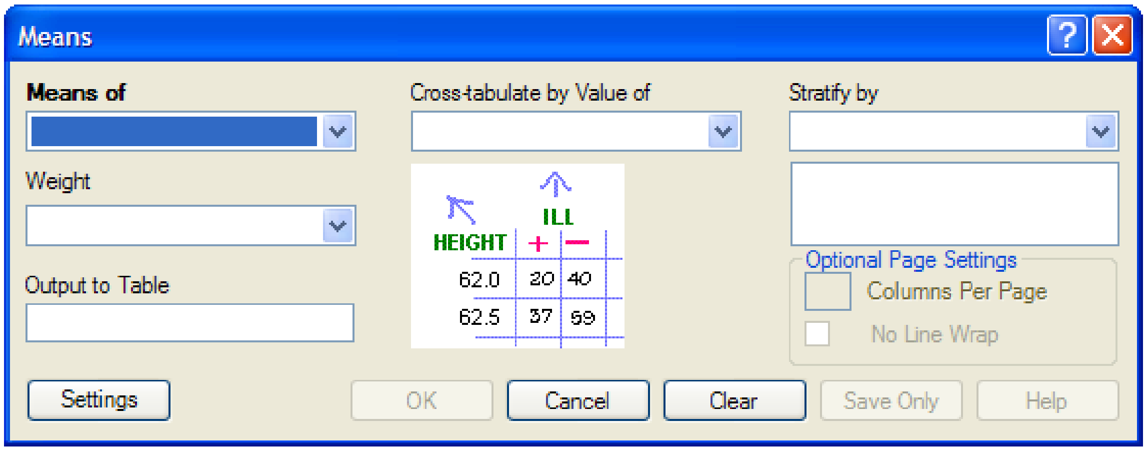


Figure 7.123: Means Dialog Box

7. From the Means Of drop-down list, select a **variable** from the data source.
    - The Cross-tabulate by value of drop-down list contains a variable to help determine if the means of a group are equal.
    - The Stratify by drop-down list contains a variable to act as a grouping variable.
    - The Weight drop-down list contains a variable for weighted Classic Analysis.

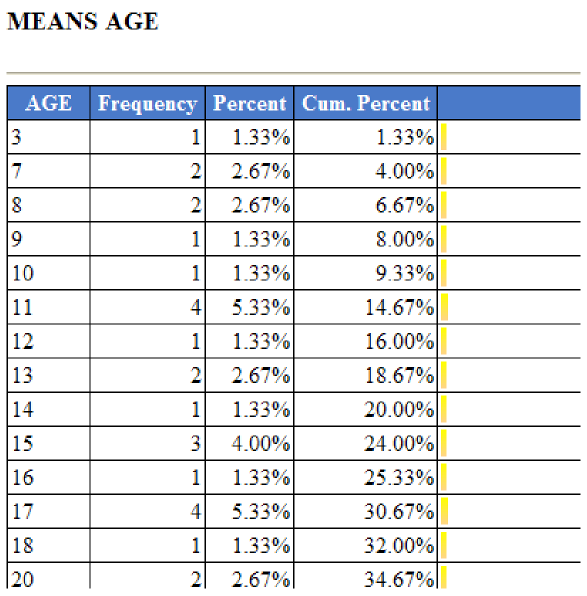
- The Output to Table field specifies a location for table results. The new table can be accessed using the READ command.

7. Click **OK**. The Output window populates with the results. Scroll down to view the Mean and Standard Deviation results.

Try It

Use the MEANS command to find the average for a continuous variable.

1. Read in the **Sample.PRJ** project. Open **Oswego**. Seventy five (75) records should be listed in the Output window.
2. Click **MEANS**. The MEANS dialog box opens.
3. From the Means of drop-down list, select **Age**.
4. Click **OK**. Results appear in the Output window.



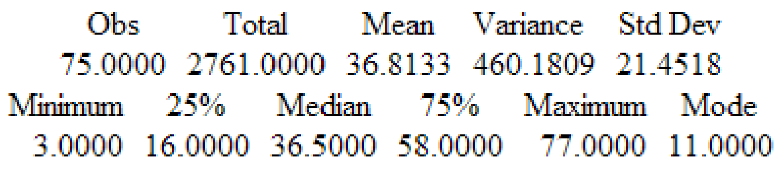
MEANS AGE

AGE	Frequency	Percent	Cum. Percent
3	1	1.33%	1.33%
7	2	2.67%	4.00%
8	2	2.67%	6.67%
9	1	1.33%	8.00%
10	1	1.33%	9.33%
11	4	5.33%	14.67%
12	1	1.33%	16.00%
13	2	2.67%	18.67%
14	1	1.33%	20.00%
15	3	4.00%	24.00%
16	1	1.33%	25.33%
17	4	5.33%	30.67%
18	1	1.33%	32.00%
20	2	2.67%	34.67%

Figure 7.124: Means Output Window

5. To view the Descriptive Statistics, scroll down the Age listing.

- Notice the mean age of individuals in the dataset is 36.8.



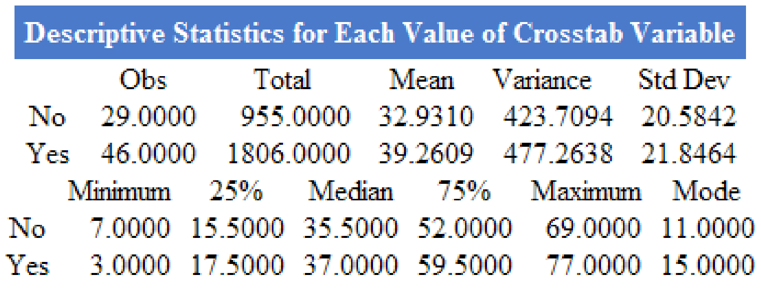
```
      Obs        Total       Mean    Variance   Std Dev
   75.0000  2761.0000  36.8133  460.1809  21.4518
Minimum     25%      Median     75%     Maximum    Mode
  3.0000  16.0000  36.5000  58.0000    77.0000  11.0000
```

Figure 7.125: Output Data

Try It

For this example, use a numeric variable and a Yes/No variable to determine the MEANS for a group.

1. Read in the **Sample.PRJ** project.
2. Click **MEANS**. The MEANS dialog box opens.
3. From the Means of drop-down list, select **Age**.
4. From the Cross-Tabulate by Value of drop-down list, select **ILL**.
5. Click **OK**. Results appear in the Output window.
6. To view the Descriptive Statistics, scroll down the Age listing.
   - Notice the Mean Age of those answering yes to ill is 39.2. Those answering no to ill is 32.9.



Descriptive Statistics for Each Value of Crosstab Variable

	Obs	Total	Mean	Variance	Std Dev
No	29.0000	955.0000	32.9310	423.7094	20.5842
Yes	46.0000	1806.0000	39.2609	477.2638	21.8464

	Minimum	25%	Median	75%	Maximum	Mode
No	7.0000	15.5000	35.5000	52.0000	69.0000	11.0000
Yes	3.0000	17.5000	37.0000	59.5000	77.0000	15.0000

Figure 7.126: Descriptive Statistics Output Window

Other available statistics include:

- ANOVA, a Parametric Test for Inequality of Population Means.
- Bartlett's Test for Inequality of Population Variances.
- Mann-Whitney/Wilcoxon Two-Sample Test (Kruskal-Wallis test for two groups).

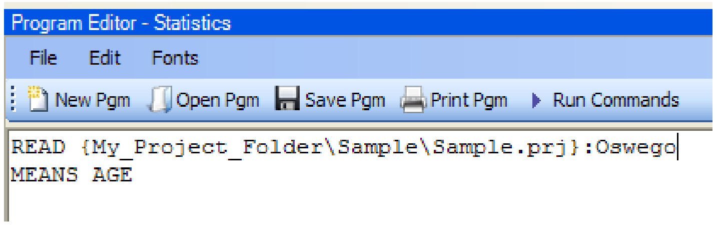
Try It

For this example, use the Classic Analysis Program Editor to create the code.

1. Read in the **Sample.PRJ** project. Open **Oswego**.

2. In the Program Editor window, place the cursor under the Read command created. Type **MEANS Age**.

   - Do **not** press the Enter key. Leave the cursor on the line of code just created.



```
READ {My_Project_Folder\Sample\Sample.prj}:Oswego
MEANS AGE
```

Figure 7.127: Means Age in Program Editor

3. Click **Run Commands** from the Program Editor navigation menu. The code turns gray to indicate execution of syntax.

   - Results of the command appear in the Output window.
   - Results are the same as those created using the Command Tree dialog boxes.

Use the SUMMARIZE Command

The SUMMARIZE command aggregates data producing an output table showing the descriptive statistics.

Syntax

```
SUMMARIZE varname::aggregate(variable) [varname::aggregate(variable) ...] TO tablename
STRATAVAR=variable list {WEIGHTVAR=variable}
```

1. From the Classic Analysis Command Tree, use the READ command to open a **PRJ project file**.
2. From the Classic Analysis Command Tree, click **Statistics > Summarize**. The Summarize dialog box opens.

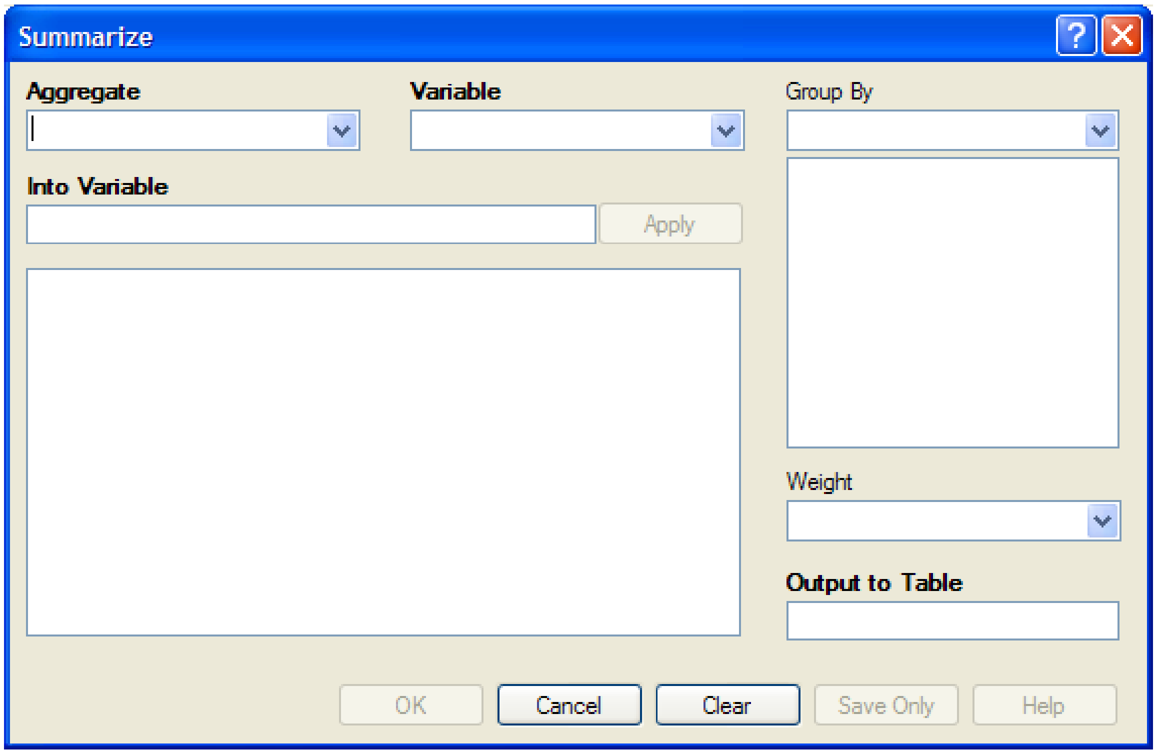


Figure 7.128: Summarize Dialog Box

3. From the Aggregate drop-down list, select from the available functions to specify how records will be combined. The options are:
   - **Average -** used only on numeric fields.

- **Count**
- **First -** based on sorts order.
- **Last -** based on sorts order.
- **Maximum**
- **Minimum**
- **StdDev -** Standard Deviation.
- **StdDev(Pop)**
- **Sum -** only used on numeric fields.
- **Var-** Variance - only used on numeric fields.
- **Var(Pop)**

4. From the Variable drop-down list, select from the **variables** in the dataset.
5. In the Into Variable field, type a **name** to title the summarized variable.
6. Click **Apply**. The code appears in the open field.
    - If grouping is necessary, use the Group By drop-down list to select a **variable** to group the aggregated data.
    - If a weight variable exists, use the Weight drop-down list to select a **variable** to weight the aggregated data.
7. In the Output to Table field, type a **name** for the new data table.
8. Click **OK**. The following message appears in the Output window: Output table created: location listed.
9. Click **Read/Import**. The READ dialog box opens.
10. Click the **All** radio button. All the tables in the dataset appear in a list.
11. Locate and select the **summary table** just created.
12. Click **OK**. The Output window populates with the Record Count for the summary table.
13. Use the LIST command to view the summary table.

Try It

1. Read in the **Sample.PRJ** project. Open **ADDFull**. Three hundred and fifty nine (359) records should be listed in the Output window.
2. Click **Statistics > Summarize**. The SUMMARIZE dialog box opens.

3. From the Aggregate drop-down list, select **Average**.
4. From the Variable drop-down list, select the variable **GPA**.
5. In the Into Variable field, type **AverageGPA**.
6. Click **Apply**. The code appears in the open field.
7. From the Group By drop-down list, select the variable **GENDER**.
8. In the Output to Table field, type **DATATABLE1**.
9. Click **OK**. The following message appears in the Output window:

   ```
   SUMMARIZE AverageGPA :: Avg(GPA) TO DATATABLE1 STRATAVAR = GENDER
   ```

10. Click **Read/Import**. The READ dialog box opens.
11. Click the **Tables** checkbox. All the tables in your dataset appear in a list.
12. Select the summary table **DATATABLE1**.
13. Click **OK**. The Output window populates with a record count of two for the summary table.
14. Use the LIST command to view the summary table.

Use the GRAPH Command

The following steps create a basic graph containing one main variable. The GRAPH command opens the Epi Graph application and creates a variety of graph types based on the types of data available in the project.

Syntax

```
GRAPH [<Variable(s)>] GRAPHTYPE ="<GraphType>" [<OptionName>=OptionValue]

GRAPH [<Variable(s)>] * <Crosstab> GRAPHTYPE ="<GraphType>" [<OptionName>=OptionValue]

GRAPH [<Variable(s)>] GRAPHTYPE="<GraphType>" XTITLE="<string>" YTITLE="<string>"
```

1. From the Classic Analysis Command Tree, use the READ command to open an Epi Info 7 **.PRJ file**.
2. From the Classic Analysis Command Tree, click **Statistics > Graph**. The GRAPH dialog box opens.
   - The default graph selection is Bar.

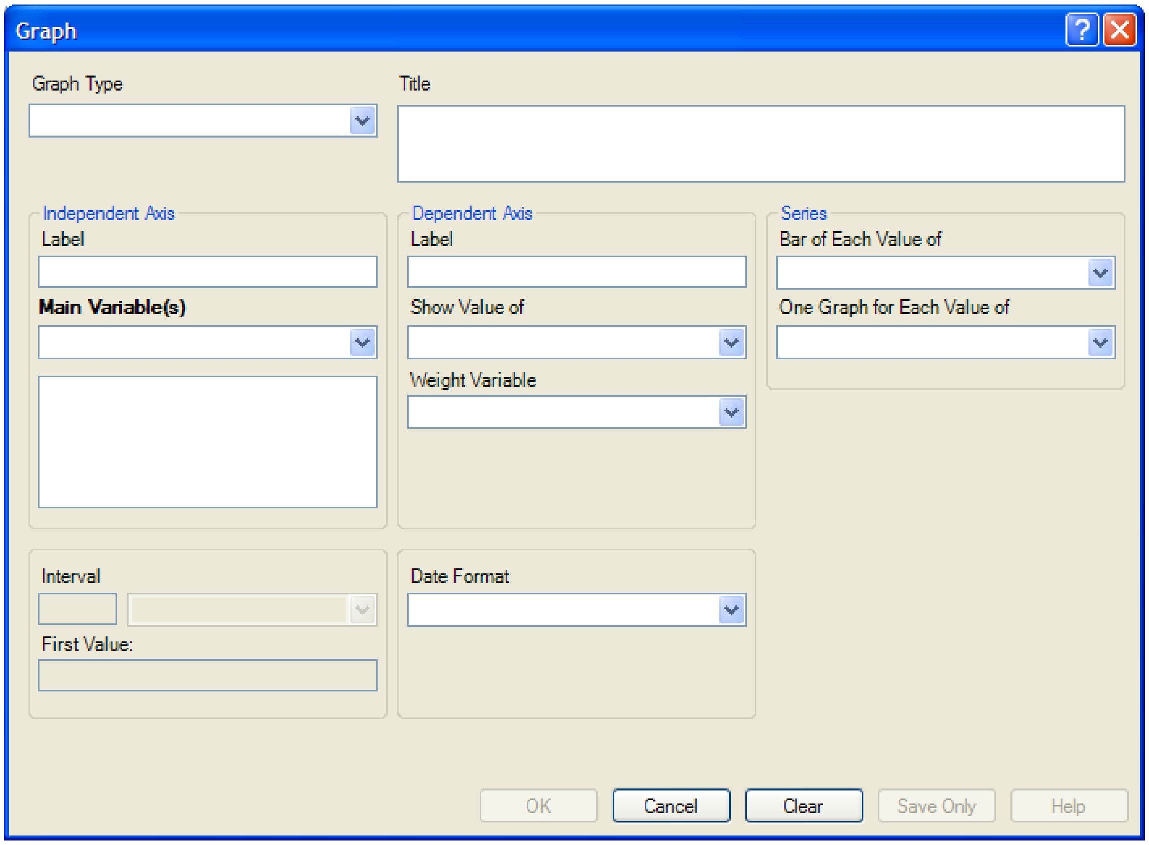


Figure 7.129: Graph Dialog Box

3. From the Graph Type drop-down list, select a type of **graph** from the list.
4. From the Main Variable(s) drop-down list, select the **X-Axis value** to graph. Use the:
    - Show Value Of drop-down list to select an **aggregate** when multiple records need to be displayed as a group.
    - Weight Variable drop-down list to select a **variable** for weighted Classic Analysis.
    - Bar for Each Value Of drop-down list to select a **variable** to generate multiple series from a single data source, each with a different color or style.
    - One Graph for Each Value of drop-down list to select a **variable** to generate a multiple series from a single data source, each displayed on a separate set of axes.
    - X-Axis Label box to type a **label** to appear in the graph.
5. Click **OK**. Epi Graph opens.
6. View the graph. Graphs can be saved by clicking on the floppy disk icon located on the top right hand side of the graph.
7. The graph is now embedded in the Output window as part of your output file.

Try It

Create a bar graph using data from the Sample.MDB project.

6. Read in the **Sample.PRJ** project. Open **Oswego**. Seventy five (75) records should be listed in the Output window.
7. Click **Statistics > Graph**. The GRAPH dialog box opens.
8. From the Graph Type drop-down list, select **Column**.
9. From the Main Variable drop-down list, select the variable **Age** to use for the X-Axis.
10. Click **OK**. Epi Graph opens.

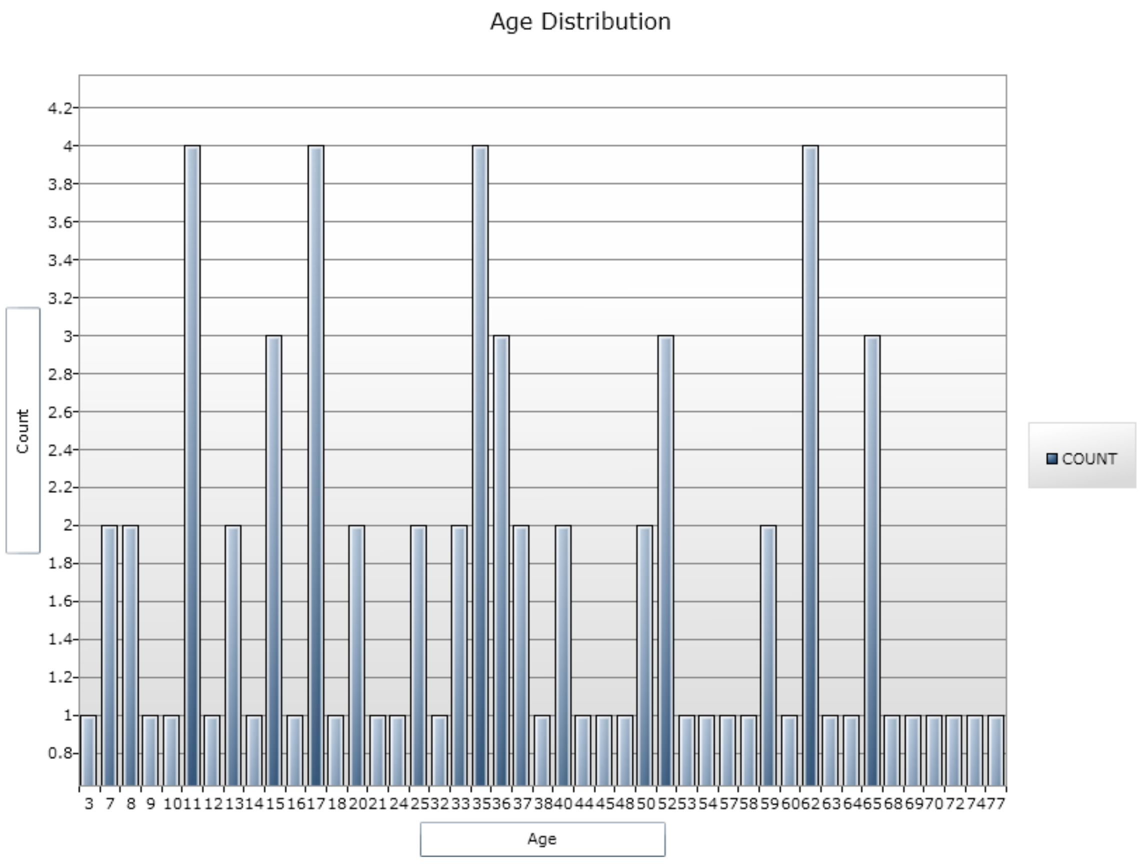


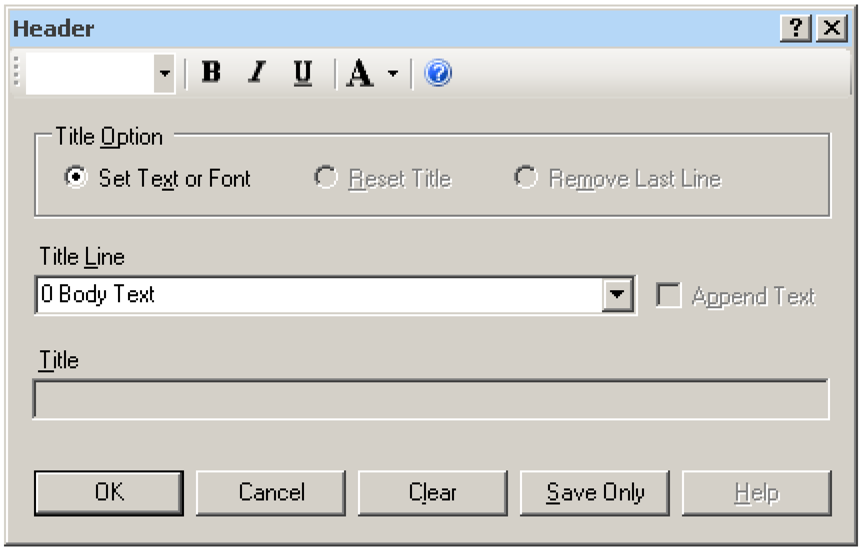
Figure 7.130: Bar Graph

6. The bar graph is now embedded in the Output window and part of the output file.
   - Notice the Graph code that appears in the Program Editor.

How to Manage Output

Header

The Header command in Classic Analysis sets up specific headings as part of the output in Analysis.



Title Option: [✓ Set Text or Font] Reset Title Remove Last Line

Title Line: 0 Body Text

Append Text

Title

Figure 7.131: Header Command Window

- **Title Option** permits title lines to be set or changed. If you select Reset Title, the selected title level will be reset to default titles. If you select Remove Last Line, the last line of a title level will be removed.
- **Title Line** indicates which level of titles or body text is affected by the command and settings.
- **Append Text** adds a new line of text to an existing header. If not selected, the current title is replaced.
- **Title** indicates the text in the header title.
- **Bold** displays header fonts in bold style text.
- **Italic** italicizes header fonts.
- **Underline** underlines header fonts.
- **Font Size** sets the text font size.
- **Font Color** allows you to apply a color to the header text.

- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Save Only** saves the created code to the Program Editor, but does not run the code.
- **Cancel** closes the dialog box without saving or executing a command.
- **Clear** empties the fields so information can be re-entered.
- **Help** opens the Help topic associated with the module being used. (Currently Disabled).

TypeOut

The TypeOut command in Classic Analysis inserts text, either a string or the contents of a file, into the output. Typical uses may include comments or boilerplates.

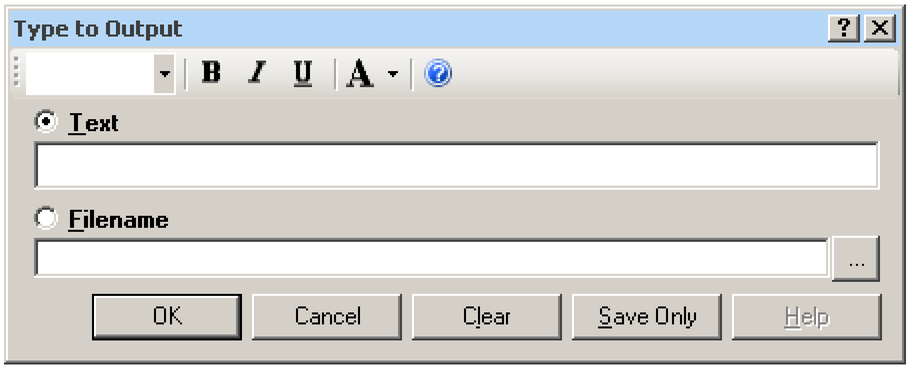
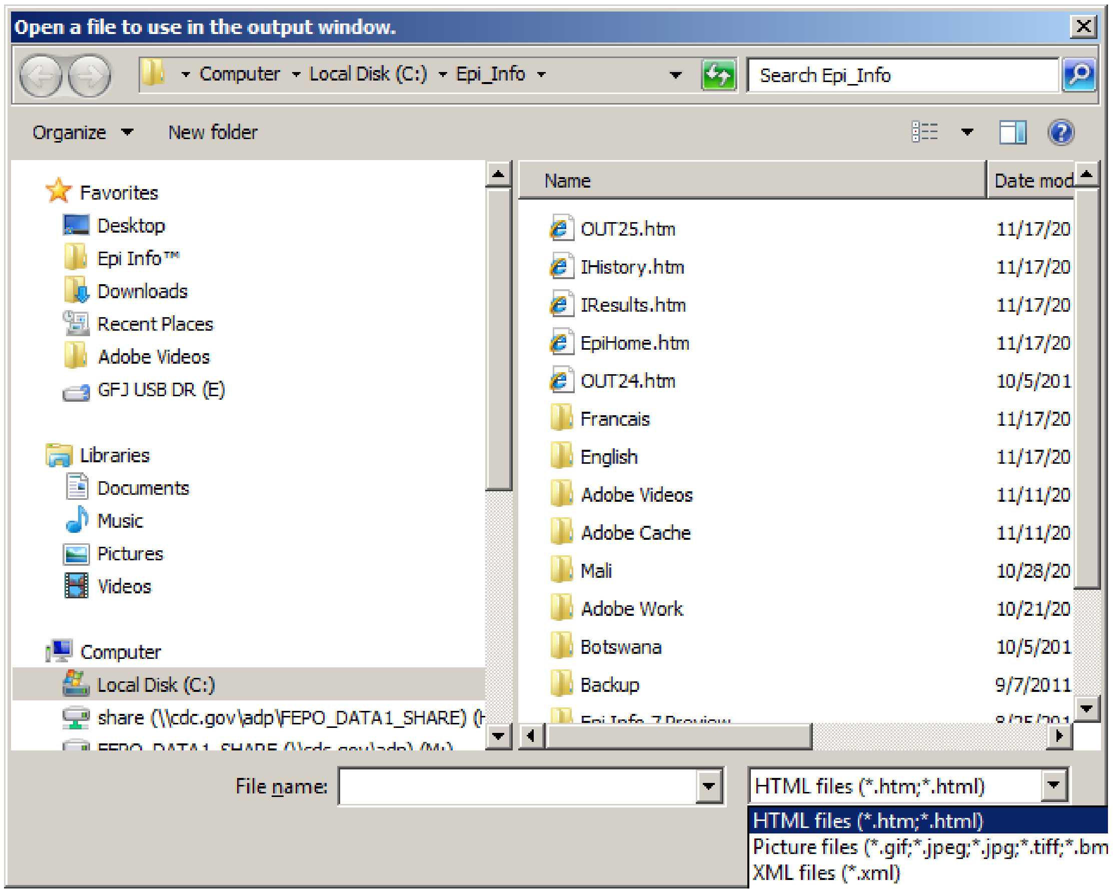


Figure 7.132: Type to Output, or TypeOut, Window

Selecting “Filename” opens the Windows File Explorer to allow you to select a file to be included in your output.



Name	Date mod
OUT25.htm	11/17/20
IHistory.htm	11/17/20
IResults.htm	11/17/20
EpiHome.htm	11/17/20
OUT24.htm	10/5/201
Francais	11/17/20
English	11/17/20
Adobe Videos	11/11/20
Adobe Cache	11/11/20
Mali	10/28/20
Adobe Work	10/21/20
Botswana	10/5/201
Backup	9/7/2011

Figure 7.133: Open File Window

- The **Text or Filename** field contains the text or file name to be inserted in the output file.
- **Bold** displays type fonts in bold style text.
- **Italic** italicizes type fonts.
- **Underline** underlines type fonts.
- **Font Size** sets the type text font size.
- **Font Color** allows you to apply a color to the type text.
- Type a **filename** or click the **ellipse** to locate a file (e.g., HTM, JPEG, XML**)**.
- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Save Only** saves the created code to the Program Editor, but does not run the code.
- **Cancel** closes the dialog box without saving or executing a command.
- **Clear** empties the fields so information can be re-entered.
- **Help** opens the Help topic associated with the module being used. (Currently Disabled).

Use the ROUTEOUT Command

The ROUTEOUT command enables you to locate output. If no directory exists, the file is placed in the current project's directory. Results accumulate until a CLOSEOUT command is executed. Output files can be placed in any folder. The ROUTEOUT command selects a path and filename. If no output file is selected, Analysis uses the default value. In each folder, Analysis creates a new index table that contains links to the files created.

1. From the Analysis Command Tree, use the READ command to open a **PRJ project file**.
2. From the Analysis Command Tree, click **Output > RouteOut**. The ROUTEOUT dialog box opens.

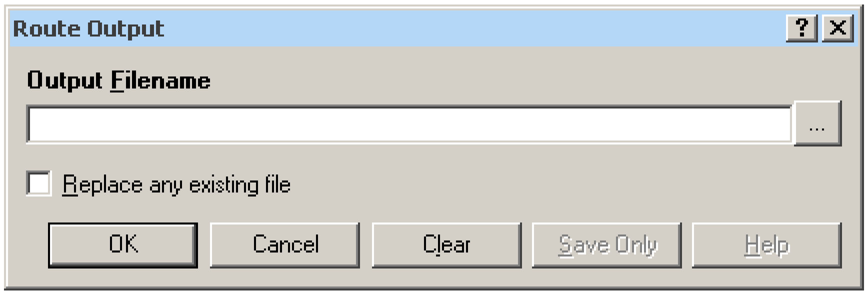


Figure 7.134: Route Output Window

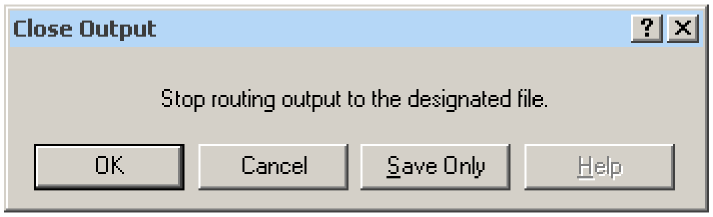
3. In the Output Filename field, enter a **file name** to locate an existing file.
   - If necessary, select **Replace Existing File** to write over any files with the same name.
4. Click **OK**. Create **Output** on the existing project.
   - Notice the title bar contains the output filename specified in the ROUTEOUT command.
   - The new file is placed in the selected directory with the extension .HTM.
5. From the Output window toolbar, click **Open**. The Browse dialog box opens.
6. Locate and select the **file** created with ROUTEOUT. Click **Open**. The Output appears in the Output window.
   - The saved Output file can be opened by any application that can read an HTML file.

To end the ROUTEOUT command, choose one of the following options.

- From the Output folder, click **Closeout**.
- READ in a **new project**.
- Close **Epi Info**.

Closeout

The Closeout command closes the current output file.



Stop routing output to the designated file.

Figure 7.135: Close Output Window

- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Save Only** saves the created code to the Program Editor, but does not run the code.
- **Cancel** closes the dialog box without saving or executing a command.
- **Help** opens the Help topic associated with the module being used. (Currently Disabled).

Printout
Command Reference

The Printout command sends the current output file, or another file specified by the user, to the default printer. This differs from the print button on the output window because a command is generated that causes printing whenever it is run.

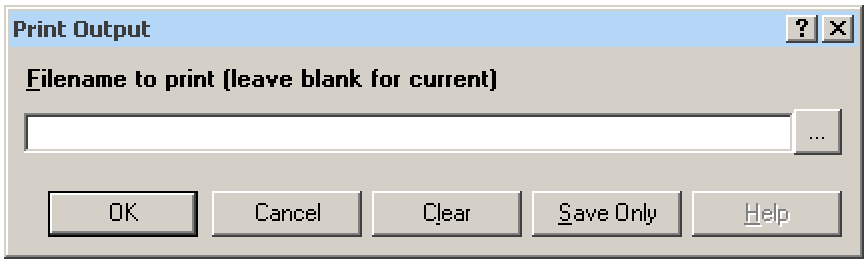


Figure 7.136: Print Output Window

- If **Filename** is selected, the file will print. If blank, it prints the current output.
- The ellipses (…) to the right of the filename opens a Windows dialog box and allows you to search the computer for the program or command to execute.
- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Save Only** saves the created code to the Program Editor, but does not run the code.
- **Cancel** closes the dialog box without saving or executing a command.
- **Clear** empties the fields so information can be re-entered.
- **Help** opens the Help topic associated with the module being used. (Currently Disabled).

Storing Output

The Storing Output command defines how output files are stored in the current project directory with a name composed of a prefix and a sequence number.

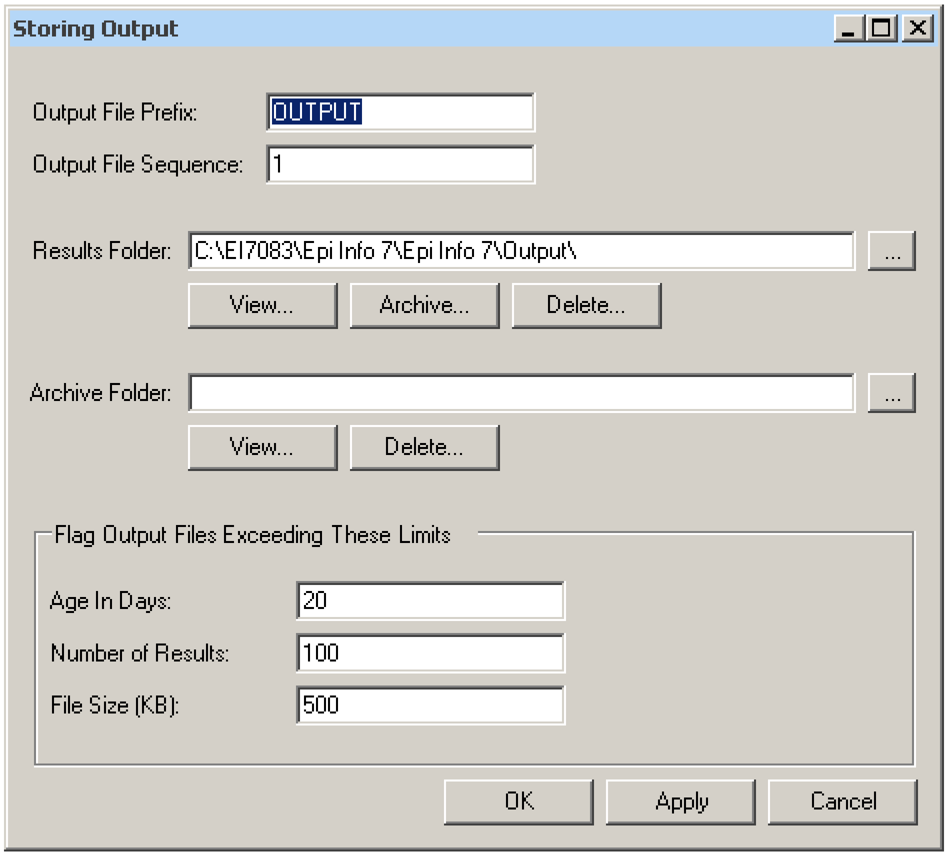


Figure 7.137: Storing Output Command Window

- **Output File Prefix** contains the first part of the filename of the output files. It is used with the Output File Sequence to store output files in the directory.
- **Output File Sequence** contains the second part of the filename for output files. This number is automatically incremented for each file. Along with the Output File Prefix, it stores output files in the directory.
- **Results Folder** locates the Results index file in the project directory.
- The **Archive Folder** allows you to locate the Archive file in the directory.
- **Archive** opens the Archive Results window, and allows specific output files to be selected and archived to the current directory.
- **Delete** opens the Delete Results window and allows you to delete specific files from the directory.

- **Age in Days** marks output files older than the indicated number of days in the form window if Flags are active. If output file limits are placed on number of days, it appears in the Flag Files operation.
- **Number of Results** marks output files in excess of the number indicated in the form window if Flags are active.
- **File Size** marks output files in excess of the size indicated in the form window if Flags are active. Enter output file size limits in this field.
- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Apply** accepts the current settings and data, and subsequently closes the form or window.
- **Cancel** exits the dialog box without saving or executing a command.

How to Use User-Defined Commands

Define Command (CMD)

The User Define Command allows you to create a block of code that can be invoked by name, like a subroutine. This is useful for sequences of commands that will be called more than once.

This feature will be available in a future version of Epi Info

Figure 7.138: Future Version Notice

- This command will be available in a future version of Epi Info 7.

Run Saved Program

Command Reference

The **Run Saved Program** command in Classic Analysis transfers control to the second program returning to the first automatically, beginning with the line following the RUN statement. A program being run from another program is similar to an INCLUDE or subroutine in other systems.

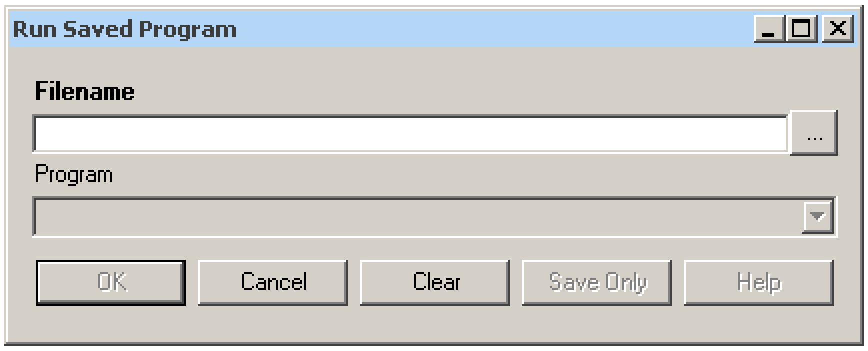


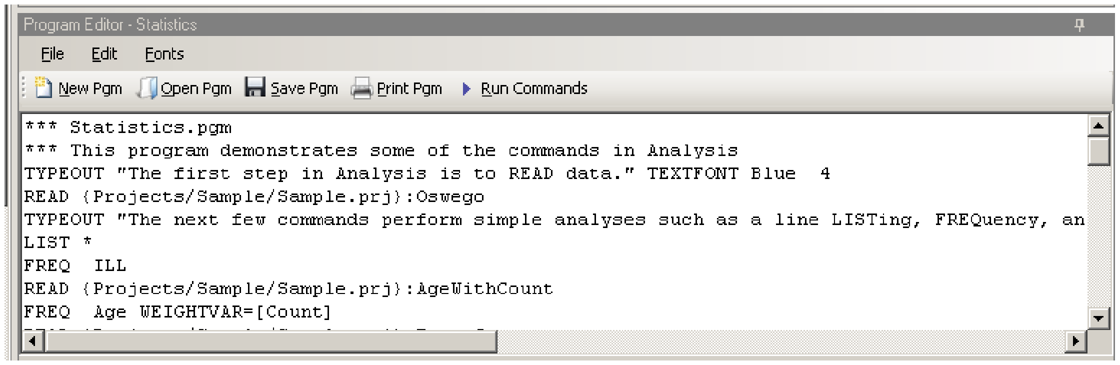
Figure 7.139: Run Saved Program

- **Filename** indicates the database or text file containing the program. If the path or filename contains a space, it must be enclosed in double quotes. If the program to be run is in the current project, the path does not need to be supplied.
- **Program** indicates the program name if it's in the database.
- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Cancel** closes the dialog box without saving or executing a command.
- **Clear** empties the fields so information can be re-entered.
- **Save Only** saves the created code to the Program Editor, but does not run the code.
- **Help** opens the Help topic associated with the module being used. (Currently Disabled).

Save and Open a Program (PGM)

Commands entered into Classic Analysis generate lines of code in the Program Editor and can be stored in the current data source (.mdb file or SQL server). Programs can be saved internally within the project or externally as a text file with a .pgm7 file extension. Saved programs that can be run as data are updated. Programs can be saved as text files, and shared without sharing data tables or projects.

Example of Code Created in the Program Editor

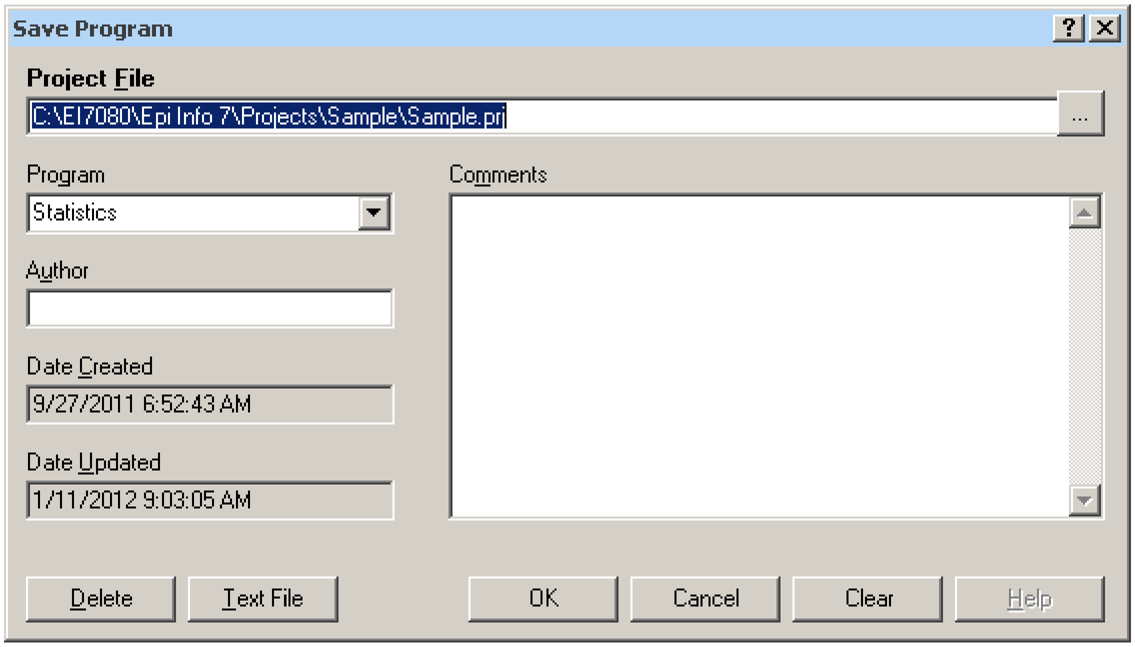


```
*** Statistics.pgm
*** This program demonstrates some of the commands in Analysis
TYPEOUT "The first step in Analysis is to READ data." TEXTFONT Blue  4
READ {Projects/Sample/Sample.prj}:Oswego
TYPEOUT "The next few commands perform simple analyses such as a line LISTing, FREQuency, an
LIST *
FREQ  ILL
READ {Projects/Sample/Sample.prj}:AgeWithCount
FREQ  Age WEIGHTVAR=[Count]
```

Figure 7.140: Example Code in Program Editor

Save Code as a PGM

1. From the Program Editor Navigation bar, click the **Save Pgm** icon. The Save Program dialog box opens.



Project File: C:\EI7080\Epi Info 7\Projects\Sample\Sample.prj

Program: Statistics

Author:

Comments:

Date Created: 9/27/2011 6:52:43 AM

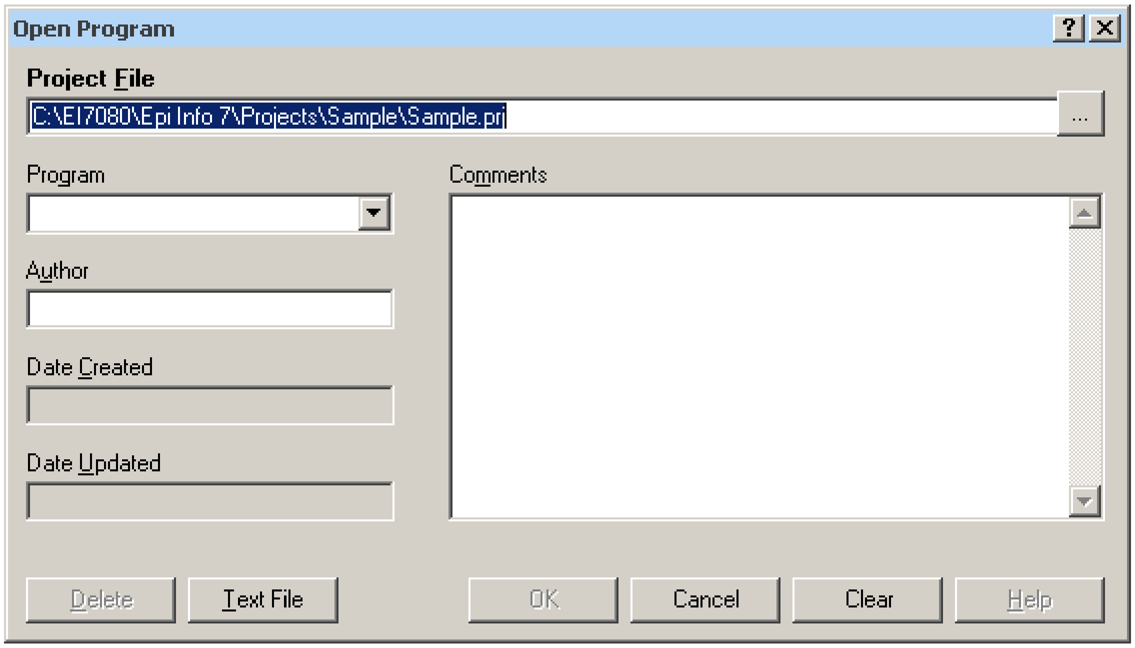
Date Updated: 1/11/2012 9:03:05 AM

Figure 7.141: Save Program Dialog Box

2. In the Program field, type a **name** for the program.
3. In the Author field, type a **name** or **initials**.
4. In the Comments field, type a **description** of the program.
5. Click **OK**.
    - When code is saved, it is written to a special table in the current .mdb, called Programs.
    - A saved program can be executed with the RUNPGM command, or opened in the Program Editor.
    - From the Save Program dialog box, click **Text File** to save code to a text file.

Open and Run a Saved PGM

1. From the Program Editor, select **File > Open Pgm** or click the **Open Pgm** icon. The Read Program dialog box opens.



Project File: C:\EI7080\Epi Info 7\Projects\Sample\Sample.prj

Figure 7.142: Read Program Dialog Box

2. From the Program drop-down list, select a saved **Pgm**.

- Information about the program automatically populates the open fields in the dialog box.
- To open a program that was saved externally, click **Text File** to view all available .pgm files.

3. Click **OK**. The program code opens in the Program Editor.
    - Code can now be run, edited, or saved.
4. Click **Run Commands**. The Program Editor runs all the code listed and displays the results in the Output window.

- To run individual commands, use the cursor to highlight the commands you want to run. Click **Run Commands**.

Execute File

Command Reference

This command executes a Windows program in Classic Analysis. You can either explicitly name the command (e.g., WinWord.exe) or one designated within the Windows registry as appropriate for a document with the named file extension (C:\Temp\MyDocument.doc). This provides a mechanism for bringing up whatever word processor or browser is the default on a computer without first knowing its name.

If the pathname is a long filename, it must be surrounded in single quotes. If the command takes parameters, surround the command and the parameters with a single set of double quotes. Do not use single quotes.

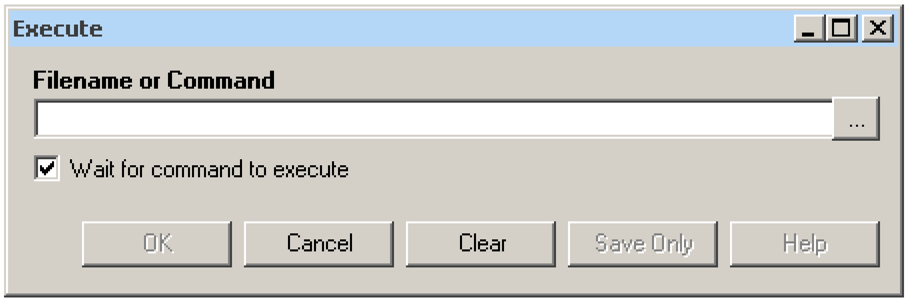


Figure 7.143: Execute File Dialog Box

- **Filename** is the file name, program name, or command to execute. Enter a path and program name for .EXE and .COM files along with any desired command line arguments.
- **Wait for command to execute** indicates whether the program should run before or after the command is executed.
- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Cancel** closes the dialog box without saving or executing a command.
- **Clear** empties the fields so information can be re-entered.
- **Save Only** saves the created code to the Program Editor, but does not run the code.
- **Help** opens the Help topic associated with the module being used. (Currently Disabled).

Syntax

```
EXECUTE <filename>

EXECUTE <program-name>
EXECUTE "<fprogram-name><command-line parameters>"

EXECUTE NOWAITFOREXIT <filename>

EXECUTE NOWAITFOREXIT '<filename>'

EXECUTE NOWAITFOREXIT "<filename>"

EXECUTE WAITFOREXIT <filename>

EXECUTE WAITFOREXIT '<filename>'

EXECUTE WAITFOREXIT "<filename>"
```

- The <program name> represents the path and program name for .exe (filename for registered Windows programs) and .com (any Internet address) files, along with any desired command line arguments. If a parameter is added, the whole string should be enclosed within double quotes.
- If EXECUTE is run modally, permanent variables are written before the command is executed, and reloaded after it is executed.

Comments

If the given name is not a program, but a file with an extension (the three characters after the ".") registered by Windows for displaying the document, the correct program to display the file will be activated (i.e., WRITEUP.DOC might cause Microsoft Word to run and load the file on one computer). Usually .TXT will run NOTEPAD.EXE or WORDPAD.EXE, and image files will appear either in a browser or graphics program. An .HTM file will bring up the default browser.

Example

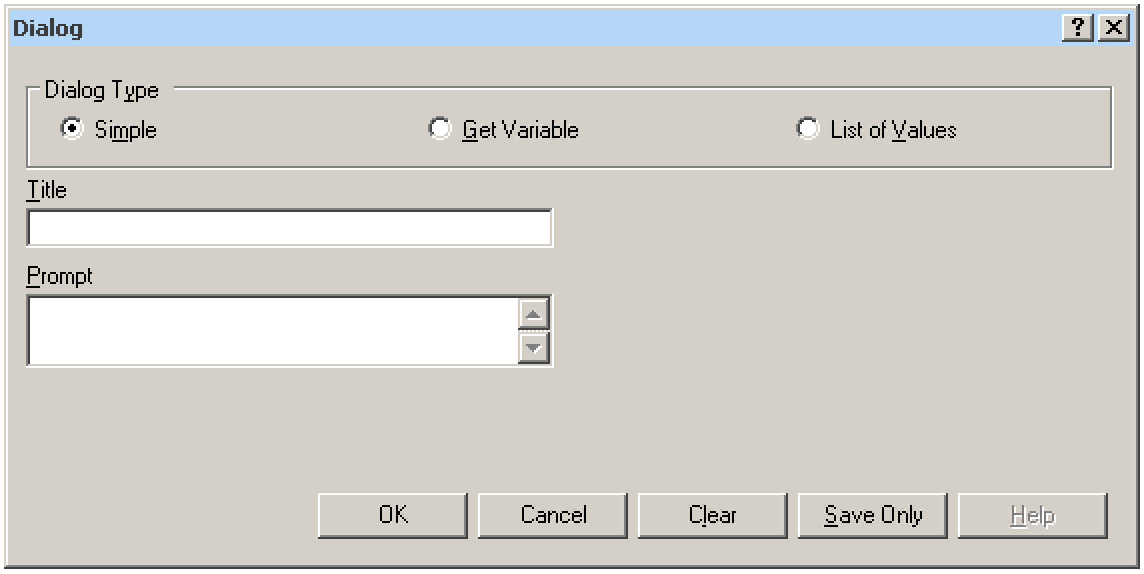
```
EXECUTE "C:\Epi_Info_7\Enter.exe sample.prj:oswego"

EXECUTE "C:\ Epi_Info_7\OUT120.htm'

EXECUTE "C:\windows\notepad.exe c:\Test1.txt"
```

How to Create User Interaction

DIALOG

How to Use the DIALOG Command

The DIALOG command provides user interaction from within a program. Dialogs can display information, ask for and receive input, and offer lists to make choices.



Dialog Type: [✓ Simple] Get Variable List of Values

Figure 7.144: Dialog Command

DIALOG Type Simple

- **Title** holds the text of the heading title.
- **Prompt** contains the text to be used in the dialog as a message.
- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Cancel** closes the command generation window without saving or executing a command.
- **Clear** empties fields so information can be re-entered.
- **Save Only** saves the command in the Program Editor, but does not run the command.
- **Help** opens the Help topic associated with the module being used. (Currently Disabled).

DIALOG Type Get Variable

This form of dialog displays a combo box of databases or form variables.

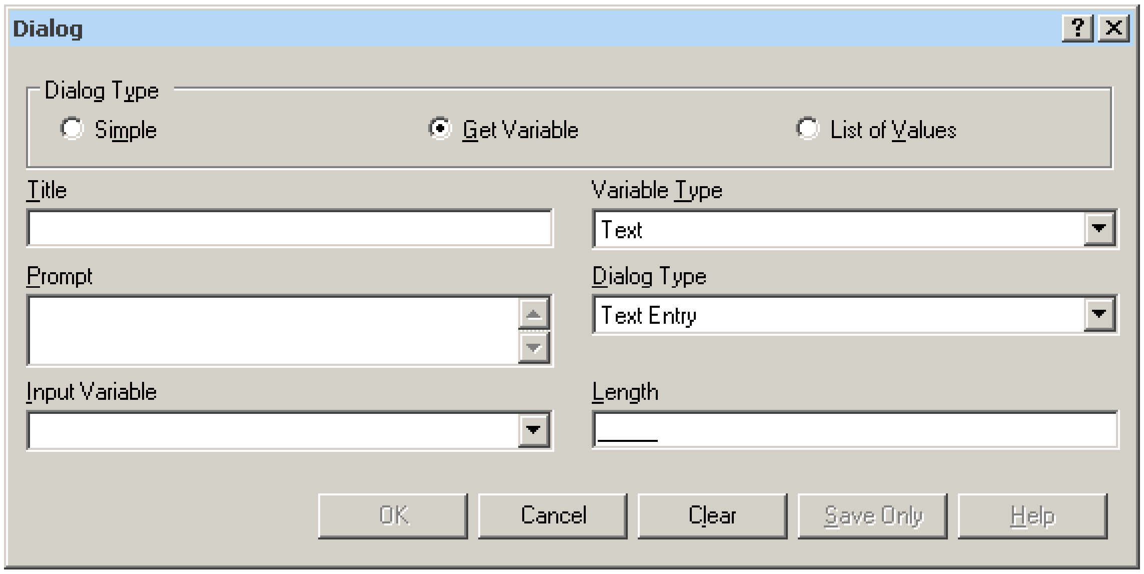


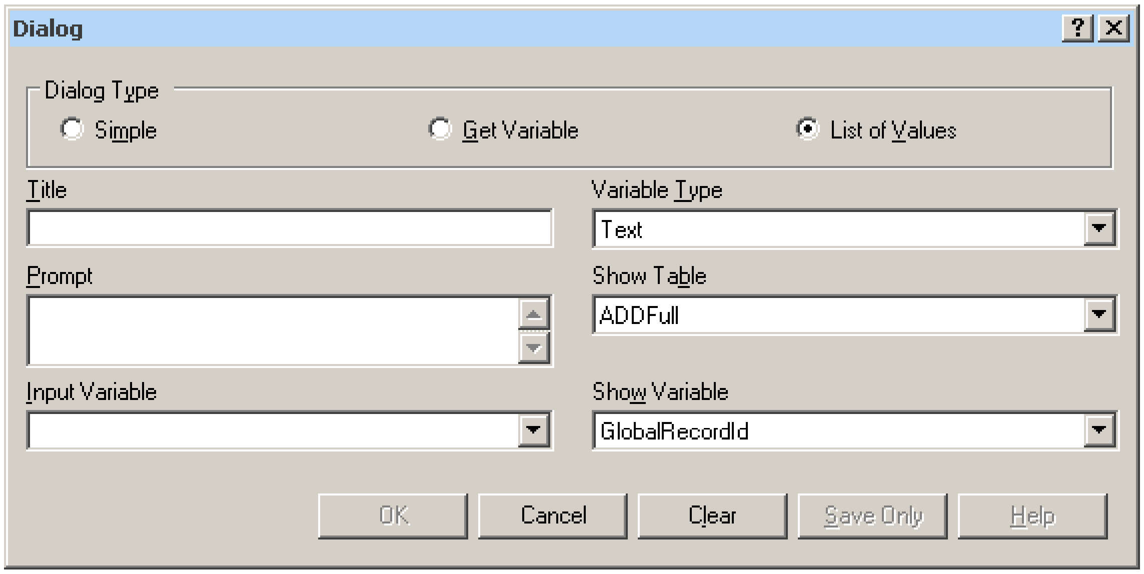
Figure 7.145: Dialog Type Combination Box

- **Title** holds the text of the heading title.
- **Prompt** contains the text to be used in the dialog as a message.
- **Input Variable** allows you to select a variable from the current data table.
- **Variable Type** allows you to select the dialog format to receive the value. Formats include Text, Number, Date, and Yes-No-Cancel. This field defaults to any previously assigned types. If the Variable Type is Text, the Dialog Type drop down allows you to select Text Entry, Multiple Choice, Variable List, Form List, Database List, File Open, and File Save. Database List options are not restricted to databases, but can include any file type. The Multiple Choice selection opens another set of dialog boxes to set up the choice selection.
- **Pattern** and/or **Length** create the pattern or size that allows input to follow when entered into the variable. Pattern options are based on Variable and Dialog Type selections. Number Type patterns consist of pound signs (#) and can contain one decimal place with a boundary. Date pattern options are DD-MM-YYYY, MM-DD-YYYY, or YYYY-MM-DD.
- **OK** accepts the current settings and data, and subsequently closes the form or window.

- **Cancel** closes the command generation window without saving or executing a command.
- **Clear** empties fields to re-enter information.
- **Save Only** saves the command in the Program Editor, but does not run the command.
- **Help** opens the Help topic associated with the module being used (Currently Disabled).

DIALOG Type List of Values

This dialog type displays a combo box of distinct values of the specified variable in the specified database.



Dialog Type: Simple | Get Variable | [✓ List of Values]

Title:

Prompt:

Input Variable:

Variable Type: Text

Show Table: ADDFull

Show Variable: GlobalRecordId

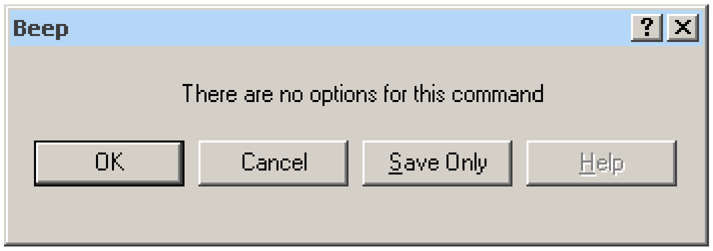
Figure 7.146: Dialog Type List of Values

- **Title** holds the text of the heading title.
- **Prompt** contains the text to be used in the dialog as a message.
- **Input Variable** allows you to select a variable from the current data table.
- **Variable Type** allows you to select the dialog format to receive the value. Options are Text, Number, Date, and Yes-No-Cancel.
- **Show Table** contains a list of available forms or tables in the current project.

- **Show Variable** contains a list of available fields that act as selection criteria for the input variable in the current project.
- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Cancel** closes the command generation window without saving or executing a command.
- **Clear** empties fields to re-enter information.
- **Save Only** saves the command in the Program Editor, but does not run the command.
- **Help** opens the Help topic associated with the module being used (Currently Disabled).

BEEP

The Beep command generates a sound when a function is performed.



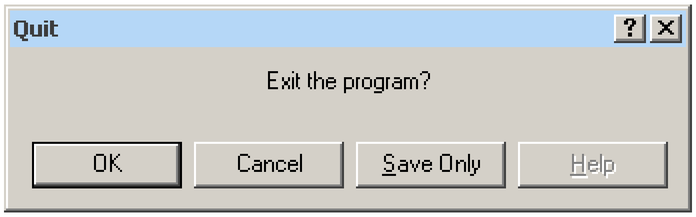
There are no options for this command

Figure 7.147: Beep Command Box

- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Cancel** closes the dialog box without saving or executing a command.
- **Save Only** saves the created code to the Program Editor, but does not run the code.
- **Help** opens the Help topic associated with the module being used (Currently Disabled).

QUIT

The Quit command closes the current data files and terminates the current program, closing Analysis.


Quit

Exit the program?

Figure 7.148: Quit Command Box

- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Cancel** closes the dialog box without saving or executing a command.
- **Save Only** saves the created code to the Program Editor, but does not run the code.
- **Help** opens the Help topic associated with the module being used (Currently Disabled).

Use IF Statements with Permanent or Global Variables

- If an IF statement condition depends on global or permanent variables, the value is computed immediately and only the true or false branch, as appropriate, is followed.
- If an IF statement condition depends on a field variable, neither branch is followed. However, computations within the branches are saved for execution, as appropriate, on each record. In the second case, non-computational commands (e.g., READ, SELECT, SORT, HEADER, and ROUTEOUT) are never executed.

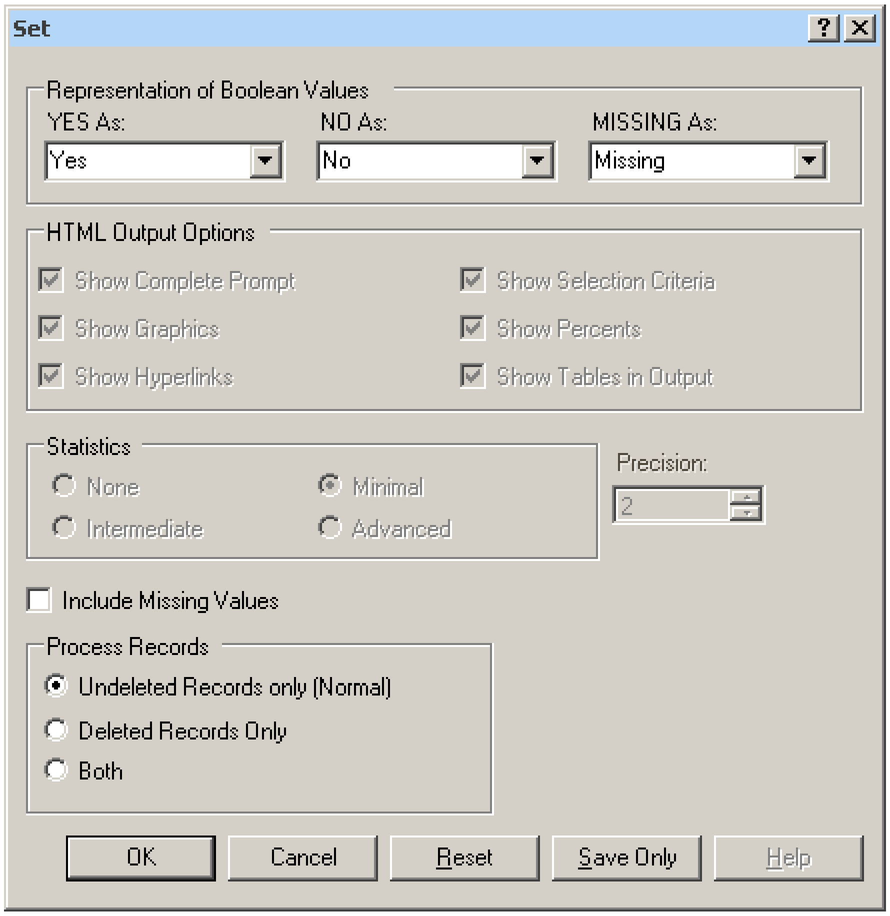
SET

Description

This command provides various options that affect the performance and output of data in Classic Analysis. These settings are utilized whenever the Classic Analysis program is used.

Syntax

```
SET [<parameter> = <value>]
```



Set

Representation of Boolean Values

YES As: Yes

NO As: No

MISSING As: Missing

HTML Output Options

[✓] Show Complete Prompt

[✓] Show Selection Criteria

[✓] Show Graphics

[✓] Show Percents

[✓] Show Hyperlinks

[✓] Show Tables in Output

Statistics

None [✓ Minimal] Intermediate Advanced

Precision: 2

[ ] Include Missing Values

Process Records

[✓ Undeleted Records only (Normal)] Deleted Records Only Both

Figure 7.149: Set Options Window

- **Representation of Boolean Values** allows you to determine how values in the Epi Info 7 projects line list are displayed for Yes/No and Checkbox fields. Boolean Values are stored in the database according to the convention used by that database. Epi Info 7 converts the line listing displayed in analysis according to the selection in this menu.

- **HTML Output Options** are not available in this version of Epi Info 7.
- **Statistics** settings are not available in this version of Epi Info 7. The grayed-out display indicates analysis statistics are advanced. The None, Minimal and Intermediate settings will be available in a future release.
- **Precision** setting is not available in this version of Epi Info 7.
- **Include Missing Values** allows you to determine if you want missing values to be included if statistical calculations are made.
- **Process Records** selects how records marked for deletion (deleted) will be processed. The normal default is to process only the undeleted records.
- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Cancel** closes the dialog box without saving or executing a command.
- **Reset** clears (removes) any changes and returns all values to their default settings.
- **Save Only** saves the created code to the Program Editor, but does not run the code.
- **Help** opens the Help topic associated with the module being used. (Currently Disabled).

How to Use Advanced Statistics

Use the REGRESS Command

The REGRESS command performs linear regression and contains support for automatic dummy variables and multiple interactions.

REGRESS can be used for simple linear regression (only one independent variable), for multiple linear regression (more than one independent variable), and for quantifying the relationship between two continuous variables (correlation). Regression is used when you want to predict one dependent variable from one or more independent variables.

Syntax

```
REGRESS <dependent variable> = <independent variable(s)> [NOINTERCEPT]
[OUTTABLE=<tablename>] [WEIGHTVAR=<weight variable>] [PVALUE=<PValue>]
```

Dialog Box

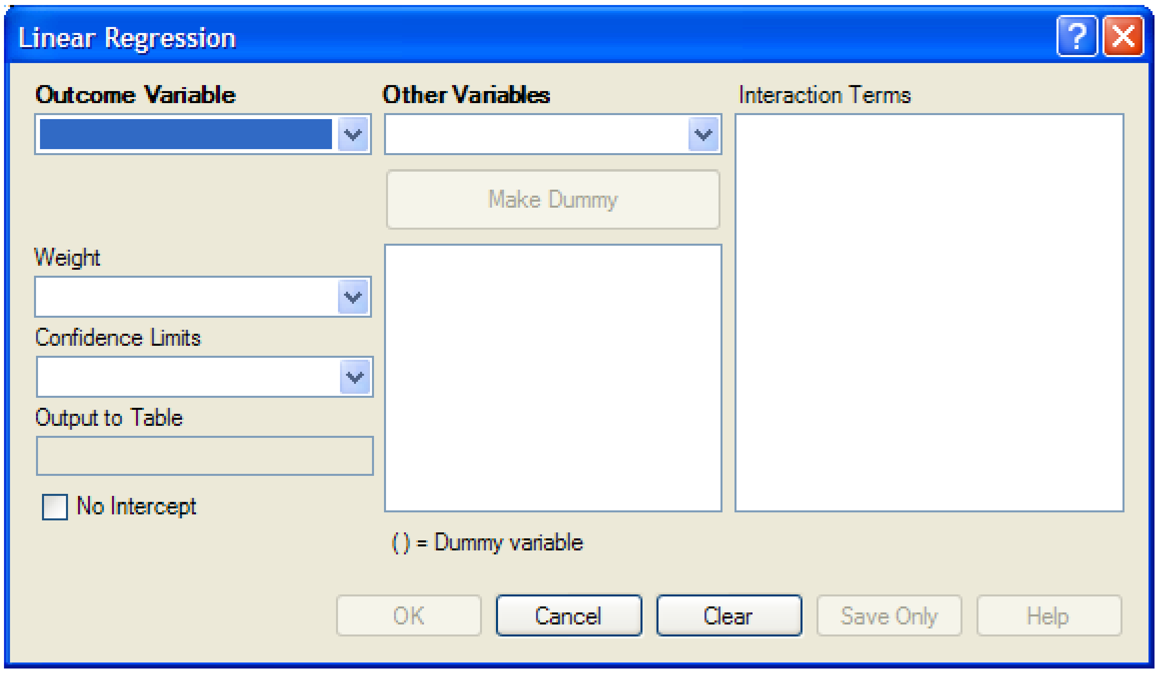


Figure 7.150: Linear Regression Window

- The **Outcome Variable** is the dependent variable for the regression.
- **Other Variables** appear in the predictor variables list.

- **Interaction Terms** are defined with the Make Interaction button. Make Interaction appears if two or more variables are selected from the Other Variables list box. If you click Make Interaction, the relationship populates the Interaction Terms list box.
- **Make Dummy** is activated if you select a predictor variable. If you select a numeric variable, it will be treated as discrete rather than continuous; the variable is enclosed in parentheses to indicate that dummy variables are being created. If a predictor variable enclosed in parentheses is selected in the list, the Make Dummy button changes to Make Continuous. Selecting it results in the variable being treated as continuous. If you select more than one predictor variable, the Make Dummy button changes to Make Interaction. Selecting it results in all possible combinations of the selected variables being added to the regression as interaction terms.
- A **Weight** variable may selected to use in weighted analyses.
- **Confidence Limits** specifies the probability level at which confidence limits are computed (default=.05).
- The **Output to Table** field identifies a table to receive output from the command. (Currently Disabled).
- If you select **No Intercept**, the regression is performed without a constant term, forcing the regression line through the origin.
- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Save Only** saves the created code to the Program Editor, but does not run the code.
- **Cancel** closes the dialog box without saving or executing a command.
- **Clear** empties the fields so information can be re-entered.
- **Help** opens the Help topic associated with the module being used. (Currently Disabled).

How to Use

7. From the Analysis Command Tree, use the READ command to open a **PRJ project file**. Select a **form** or **table**.
8. From the Analysis Command Tree, click **Advanced Statistics > Linear Regression**. The REGRESS dialog box opens.

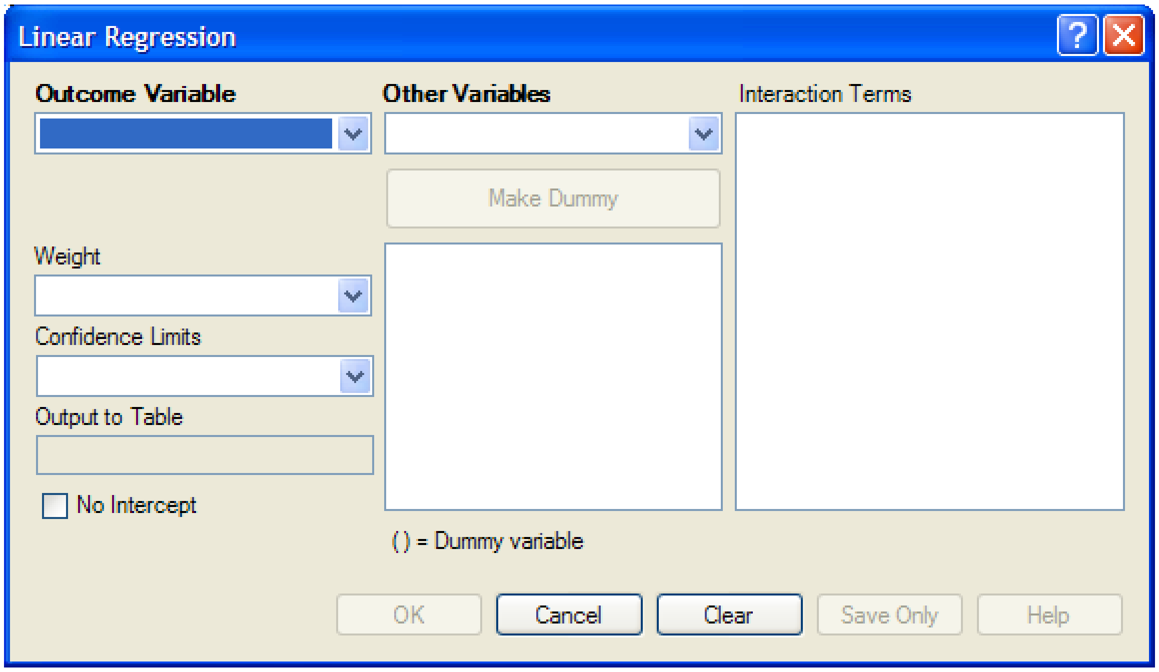


Figure 7.151: Linear Regression Window

9. From the Outcome Variable drop-down list, select a **variable** to be the dependent variable for regression.

10. From the Other Variables drop-down list, select the **variable(s)** to be the predictors.

    - If you select any predictor variables, Make Dummy will be activated. Selecting one will allow a numeric variable to become discrete rather than continuous; the variable is enclosed in parentheses to indicate that dummy variables are created. If a predictor variable enclosed in parentheses is selected, the Make Dummy button changes to Make Continuous. Selecting it makes the variable continuous. If you select more than one, the Make Dummy button changes to Make Interaction. Selecting it adds all possible selected combinations to the regression as interaction terms.

    - Make Interaction appears if you select two or more variables from the Other Variables list box. Interaction Terms are defined with the Make Interaction button. Click **Make Interaction.** The relationship populates the Interaction Terms list.

    - The Output to Table field identifies a data table to receive output from the command. Results can be sent to an output table for graphing. (Currently Disabled).

    - If No Intercept is selected, the regression is performed without a constant term, forcing the regression line through the origin.

11. Click **OK**. Results appear in the Output window.

Try It

1. Read in the project **Sample.PRJ.** Open **BabyBloodPressure**.
2. Click **Linear Regression**. The REGRESS dialog box opens.
3. From the Outcome Variable drop-down list, select **SystolicBlood**.
4. From the Other Variables drop-down list, select **AgeInDays**.
5. From the Other Variables drop-down list, select **Birthweight**.
6. Click **OK**. Results appear in the Output window.

Linear Regression

Variable	Coefficient	Std Error	F-test	P-Value
AgeInDays	5.888	0.680	74.9229	0.000002
Birthweight	0.126	0.034	13.3770	0.003281
CONSTANT	53.450	4.532	139.1042	0.000000

Correlation Coefficient: r^2= 0.88

Source	df	Sum of Squares	Mean Square	F-statistic
Regression	2	591.036	295.518	48.081
Residuals	13	79.902	6.146	
Total	15	670.938		

Use the LOGISTIC Command

The Logistic Regression command performs conditional or unconditional multivariate logistic regression with automatic dummy variables and support for multiple interactions.

The dependent (Outcome) variable must have a Yes/No value. Records with missing values are excluded from the analyses.

Independent (Other Variables) can be numeric, categorical, or Yes/No variables. Missing is interpreted as missing, 0 is false, and any other response is true. Independent variables are controlled by the Include Missing setting. If Include Missing is used with missing values and true and false, dummy variables will be made automatically, which contribute Yes vs. Missing and No vs. Missing. Independent variables of text type are automatically turned into dummy variables, which compare each value relative to the value lowest in the sort order. Date or numeric type Independent variables are treated as continuous variables unless surrounded by parentheses in the command. If that occurs, they automatically turn into dummy variables which compare each value relative to the lowest value.

Syntax

```
LOGISTIC <dependent variable> = <independent variable(s)> [MATCHVAR=<match variable>]
[NOINTERCEPT] [OUTTABLE=<tablename>] [WEIGHTVAR=<weight variable>] [PVALUE=<PValue>]
```

Dialog Box

1. From the Analysis Command Tree, use the READ command to open a **PRJ project file**. Select a **form** or **table**.
2. From the Analysis Command Tree, click **Advanced Statistics > Logistic Regression**. The LOGISTIC dialog box opens.

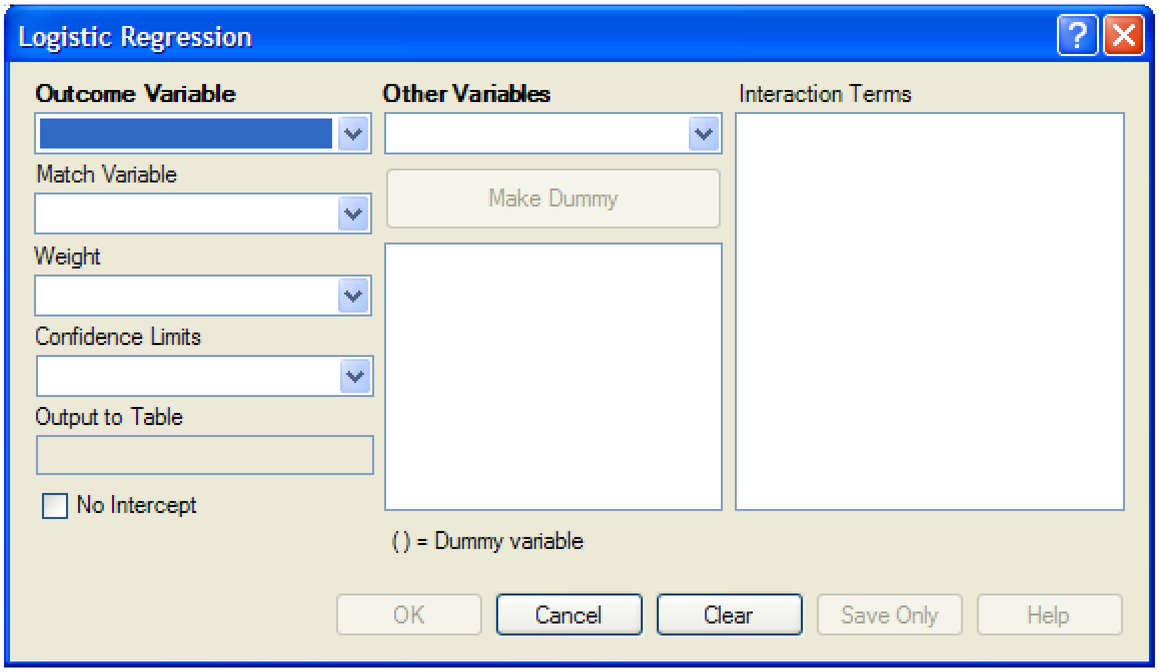


Figure 7.152: Logistic Regression Window

3. From the Outcome Variable drop-down list, select a **variable** to act as the dependent variable for regression.
4. From the Other Variables drop-down list, select the **variable(s)** to act as the predictors.
   - If you select any predictor variables, Make Dummy will be activated. Selecting one will allow a numeric variable to become discrete rather than continuous; the variable is enclosed in parentheses to indicate that dummy variables are created. If a predictor variable enclosed in parentheses is selected, the Make Dummy button changes to Make Continuous. Selecting it makes the variable continuous. If you select more than one, the Make Dummy button changes to Make Interaction. Selecting it adds all possible selected combinations to the regression as interaction terms.
   - **Make Interaction** appears if two or more variables are selected from the Other Variables list box. Interaction Terms are defined with the Make Interaction button. Click **Make Interaction.** The relationship populates the Interaction Terms list.
   - **Match Variable** identifies the variable indicating the group membership of each record.
   - The **Output to Table** field identifies a data table to receive output from the command. Results can be sent to an output table for graphing. (Currently Disabled).

- If **No Intercept** is selected, the regression is performed without a constant term forcing the regression line through the origin.

5. Click **OK**. Results appear in the Output window.

Try It

1. Read in the project **Sample.PRJ.** Open **Oswego**
2. Click **Logistic Regression**. The LOGISTIC dialog box opens.
3. From the Outcome Variable drop-down list, select **ILL**.
4. From the Other Variables drop-down list, select **BROWNBREAD**, **CABBAGESAL**, **WATER**, **MILK**, **CHOCOLATE**, and **VANILLA**.
5. Click **OK**. Results appear in the Output window.

Unconditional Logistic Regression

Term	Odds Ratio	95%	C.I.	Coefficient	S. E.	Z-Statistic	P-Value
BROWNBREAD (Yes/No)	1.7803	0.3932	8.0614	0.5768	0.7706	0.7485	0.4542
CABBAGESAL (Yes/No)	1.1342	0.2818	4.5647	0.1259	0.7104	0.1772	0.8593
WATER (Yes/No)	1.1122	0.2670	4.6326	0.1063	0.7280	0.1460	0.8839
MILK (Yes/No)	0.1342	0.0068	2.6635	-2.0086	1.5246	-1.3174	0.1877
CHOCOLATE (Yes/No)	1.0975	0.3024	3.9829	0.0930	0.6577	0.1415	0.8875
VANILLA (Yes/No)	26.0016	5.4707	123.5818	3.2582	0.7953	4.0968	0.0000
CONSTANT	*	*	*	-2.1277	0.9733	-2.1861	0.0288

Convergence:	Converged
Iterations:	5
Final -2*Log-Likelihood:	69.2504
Cases included:	74

Test	Statistic	D.F.	P-Value
Score	28.0180	6	0.0001
Likelihood Ratio	29.8484	6	0.0000

Use the KMSURVIVAL Command

The KMSURVIVAL command performs Kaplan-Meier (KM) Survival Analysis. The objective of this methodology is to estimate the probability of survival of a defined group at a designated time interval. KM uses a non-parametric survival function for a group of patients (their survival probability at time *t*) and does not make assumptions about the survival distribution.

What distinguishes survival analysis from most other statistical methods is the presence of “censoring” for incomplete observations. In a study following two different treatment regimens, analysis of the trial typically occurred well before all patients died. For those still alive at the time of analysis, the true survival time was known only to be greater than the time observed to date. These observations are called “censored.” Two other sources of incomplete observation are patients “lost to follow-up,” and the appearance of an event other than the event being studied.

Survival analysis requires censored and time variables, the units of time, and the groups being compared.

Syntax

```
KMSURVIVAL <TimeVar> = <GroupVar> *  <CensorVar> (<Value>) [TIMEUNIT="<TimeUnit>"]
[OUTTABLE=<TableName>] [GRAPHTYPE ="<GraphType>"] [WEIGHTVAR=<WeightVar>]
```

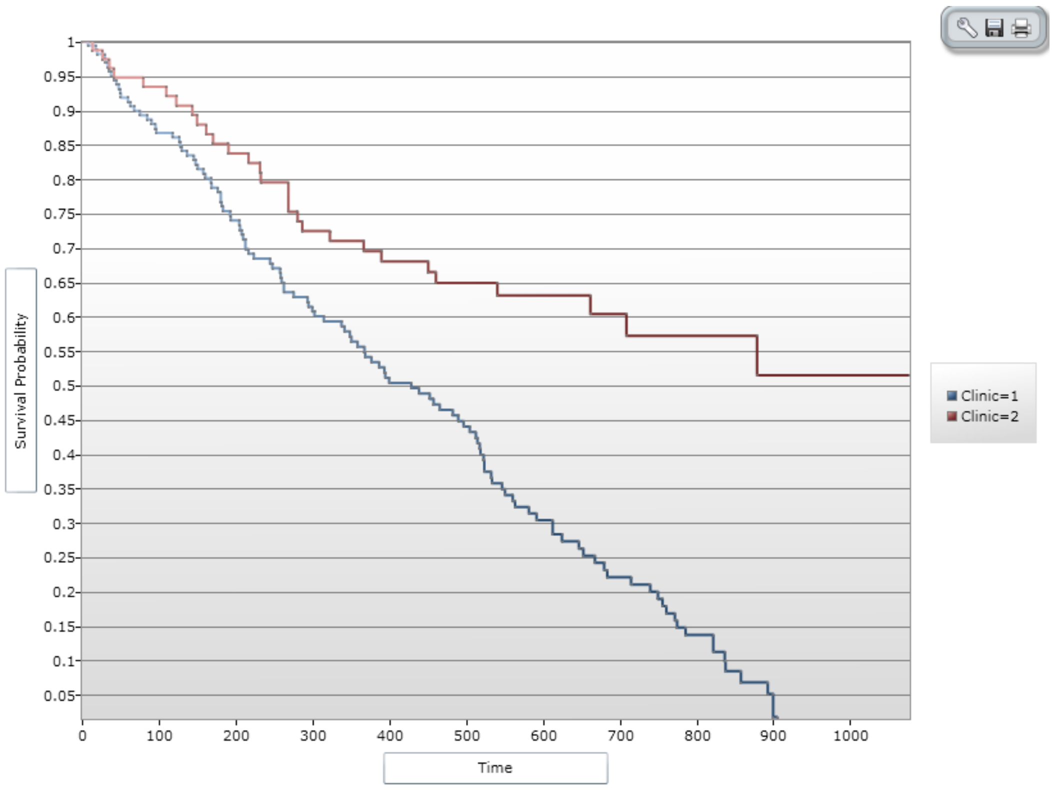
Dialog Box

1. From the Analysis Command Tree, use the READ command to open a **PRJ project file**. Select a **form** or **table**.
2. From the Analysis Command Tree, click **Advanced Statistics > Kaplan-Meier Survival**. The Kaplan-Meier Survival dialog box opens.
3. From the Censored Variable drop-down list, select the **variable** that indicates whether the case is a failure or a censored case.
4. From the Value for Uncensored drop-down list, select the **value of the censored variable** that indicates a failure**.**
5. From the Time Variable drop-down list, select the **variable** that indicates at which time the failure or censorship occurred**.**
6. From the Test Group Variable drop-down list, select the **discrete variable** used to divide cases into groups.
7. Click **OK**. Results appear in the Output window.

- A Survival Probability graph is produced by default. Use the Graph Type drop-down list to select the **Log-Survival** graph type or **None** to view the results in a table.

Try It

1. Read in the **Sample.PRJ** project. Open the **Addicts** table.
2. Click **KAPLAN-MEIER SURVIVAL**. The Kaplan-Meier Survival dialog box opens.
3. From the Censored Variable drop-down list, select **Status**.
4. From the Value for Uncensored drop-down list, select **1**.
5. From the Time Variable drop-down list, select **Survival_Time_Days**.
6. From the Test Group Variable drop-down list, select **Clinic**.
7. Click **OK**. Results appear in the Output window.



Test	Statistic	D.F.	P-Value
Log-Rank	27.2477	1	0.0000
Wilcoxon	11.6304	1	0.0007

Figure 7.153: Results of Kaplan-Meier Example Survey

Use the COXPH Command

The COXPH command performs Cox Proportional Hazards survival analysis. This form of survival analysis relates covariates to failure through hazard ratios. A covariate with a hazard ratio less than one suggests improved survival for the level being compared with the reference level. COXPH output includes regression coefficients, test statistics with p-values, hazard ratios, and cumulative survival plots.

What distinguishes survival analysis from most other statistical methods is the presence of "censoring" for incomplete observations. In a study following two different treatment regimens, analysis of the trial typically occurred well before all patients died. For those still alive at the time of analysis, the true survival time is known only to be greater than the time observed to date. These observations are called "censored". Two other sources of incomplete observation are patients "lost to follow-up", and the appearance of an event other than the event being studied.

Survival analysis requires censored and time variables, the units of time, and the groups being compared.

Syntax

```
COXPH <time variable>= <covariate(s)>[: <time function>:]  *  <censor variable>
(<value>) [TIMEUNIT="<time unit>"] [OUTTABLE=<tablename>] [GRAPHTYPE="<graph type>"]
[WEIGHTVAR=<weight variable>] [STRATAVAR=<strata variable(s)>] [GRAPH=<graph
variable(s)>]
```

Dialog Box

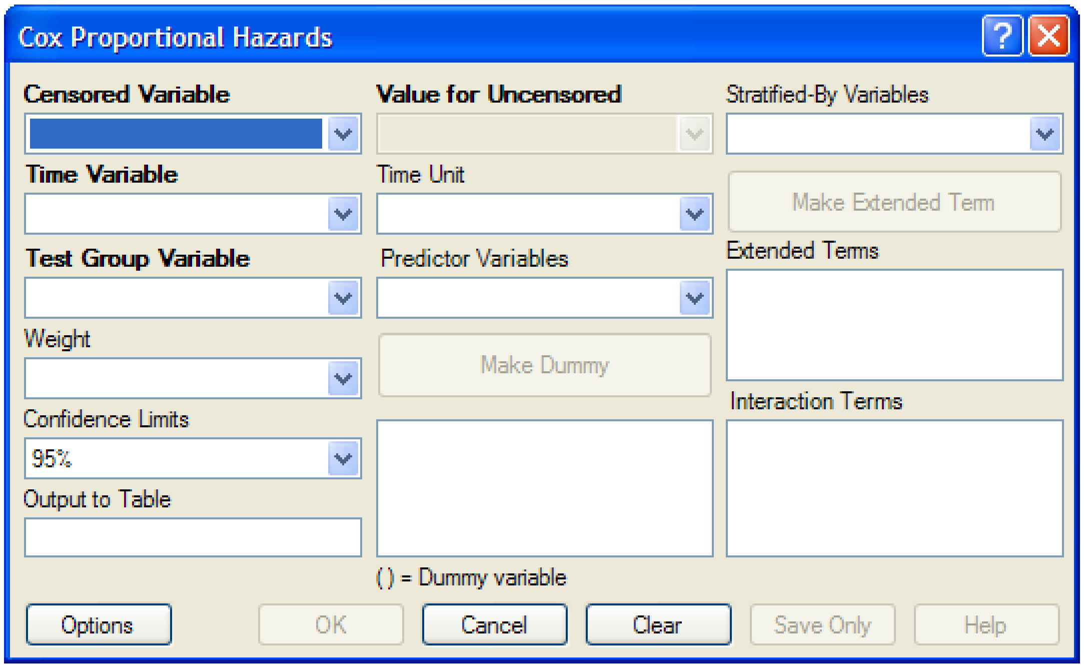


Figure 7.154: Cox Proportional Hazards Dialog Box

- The **Censored Variable** indicates whether the case is a failure or censored.
- The **Time Variable** indicates at which time the failure or censorship occurred.
- **Test Group Variable** is a discrete variable used to divide cases into groups for comparison. In the Cox model, there is no essential difference between the group variable and other predictor terms. For this reason, its use is optional.
- A **Weight** variable is selected for use in weighted analyses. The weight variable applies to all aggregation clauses.
- **Confidence Limits** indicate the probability level at which confidence limits should be computed (default=.05).
- The **Output to Table** field identifies a data table to receive output from the command. (Currently Disabled).
- **Value for Uncensored** is the value of the censored variable that indicates a failure.
- **Time Unit** is the unit in which the time variable is expressed.
- **Predictor Variables** populates the predictor variables list.
- If a predictor variable is selected, **Make Dummy** is active. If selected, a numeric variable is treated as discrete rather than continuous; the variable is enclosed in parentheses to indicate that dummy variables are being created. If you select a predictor variable enclosed in parentheses, the button changes to Make Continuous. Selecting it results in the variable being treated as continuous.
- **Stratify by** identifies the variable (if any) to be used to stratify data.
- **Options** opens the graph selection window and allows you to select variables for graphing and graph types.
- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Save Only** saves the created code to the Program Editor, but does not run the code.
- **Cancel** closes the dialog box without saving or executing a command.
- **Clear** empties the fields so information can be re-entered.
- **Help** opens the Help topic associated with the module being used. (Currently Disabled).

Try It

1. From the Analysis Command Tree, use the READ command to open a **PRJ project file**. Select a **form** or **table**.
2. From the Analysis Command Tree, click **Advanced Statistics > Cox Proportional Hazards**. The Cox Proportional Hazards dialog box opens.
3. From the Censored Variable drop-down list, select the **variable** that indicates whether the case is a failure or a censored case.
4. From the Value for Uncensored drop-down list, select the **value** of the censored variable that indicates a failure.
5. From the Time Variable drop-down list, select the **variable** that indicates at which time the failure or censorship occurred.
6. From the Time Unit drop-down list, select the **unit** in which the time variable is expressed, if needed.
7. From the Test Group Variable drop-down list, select the **discrete variable** used to divide cases into groups.
   - In the Cox model, there is no essential difference between the group variable and other predictor terms. For this reason, using the group variable is optional.
8. From the Other Variables drop-down list, select the **predictor variables.**
   - If you select any predictor variables, Make Dummy will be activated. Selecting will allow a numeric variable to become discrete rather than continuous; the variable is enclosed in parentheses to indicate that dummy variables are created. If a predictor variable enclosed in parentheses is selected in the list, the Make Dummy button changes to Make Continuous. Selecting it makes the variable continuous. If you select more than one, the Make Dummy button changes to Make Interaction. Selecting it results in all possible combinations of the selected variables being added to the regression as interaction terms.
9. Click **Graph Options**. The Cox Graph Options dialog box opens.
10. From the Plot Variables drop-down list, select a **graph type.**
11. From the list box, select a **variable(s) to graph**.
12. Clear the **Customize Graph** checkbox.
13. Click **OK**. The Cox Proportional Hazards dialog box opens.
14. Click **OK**. Results appear in the Output window.

Try It

1. Read in the **Sample.PRJ** project. Open the **Addicts** table.
2. Click **Cox Proportional Hazards**. The Cox Proportional Hazards dialog box opens.

3. From the Censored Variable drop-down list, select **Status**.
4. From the Value for Uncensored drop-down list, select **1**.
5. From the Time Variable drop-down list, select **Survival_Time_Days**.
6. From the Test Group Variable drop-down list, select **Clinic**.
7. From the Predictor Variables drop-down list, select **Methadone_dose__mg_day** and **Prison_Record**.
8. Click **Options**. The Cox Graph Options dialog box opens.
9. From the Plot Variables drop-down list, select **Survival Observed**.
10. From the list box, select **Clinic**.
11. Clear the **Customize Graph** checkbox.
12. Click **OK**. The Cox Proportional Hazards dialog box opens.
13. Click **OK**. Results appear in the Output window.

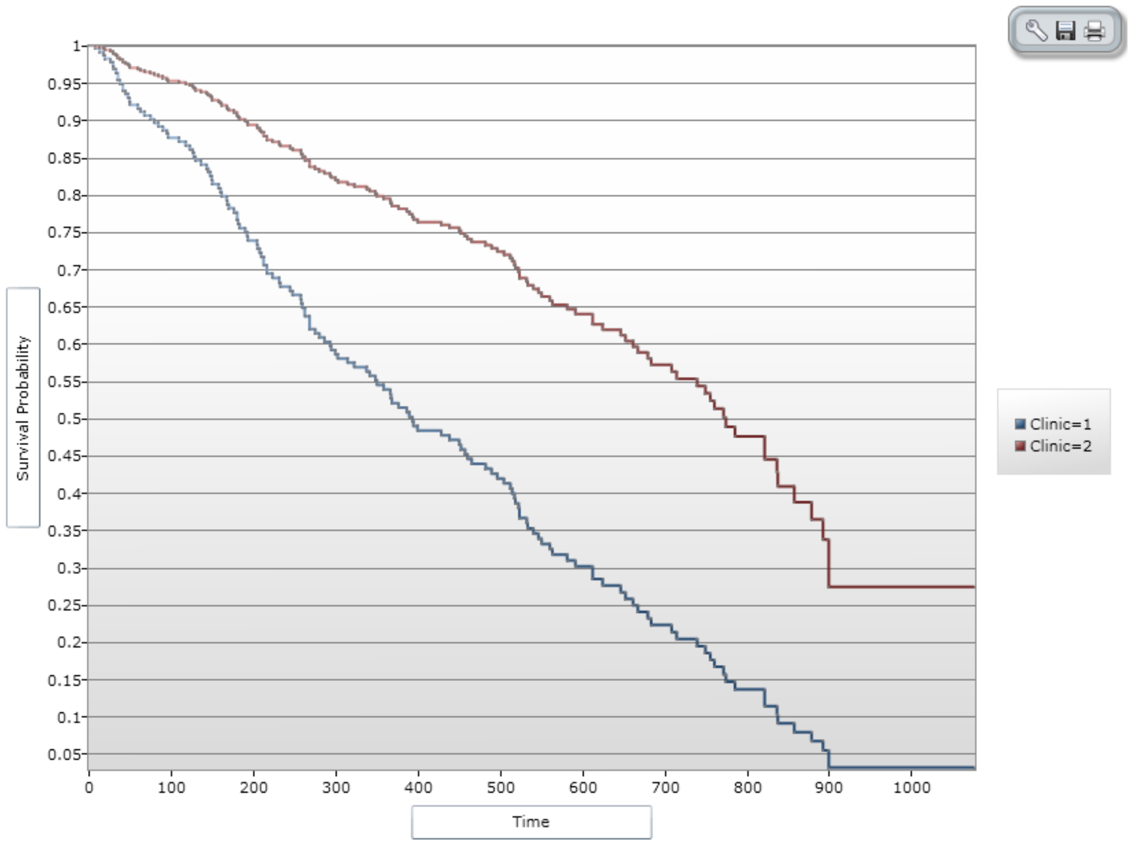
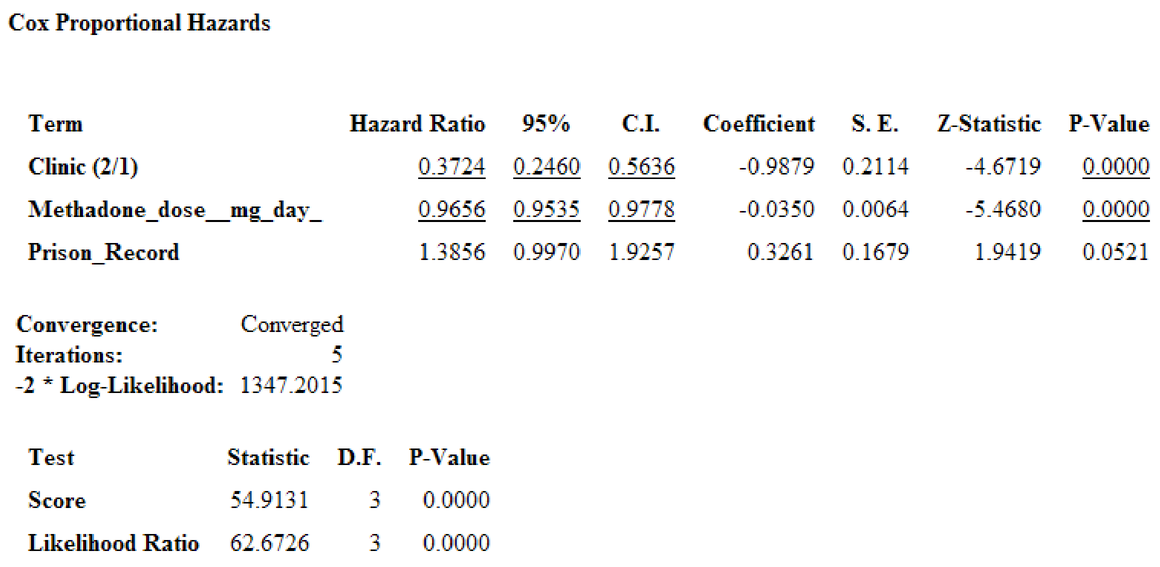


Figure 7.155: Cox Proportional Hazards Graph Result



Cox Proportional Hazards

Term	Hazard Ratio	95%	C.I.	Coefficient	S. E.	Z-Statistic	P-Value
Clinic (2/1)	0.3724	0.2460	0.5636	-0.9879	0.2114	-4.6719	0.0000
Methadone_dose__mg_day_	0.9656	0.9535	0.9778	-0.0350	0.0064	-5.4680	0.0000
Prison_Record	1.3856	0.9970	1.9257	0.3261	0.1679	1.9419	0.0521

Convergence:	Converged
Iterations:	5
-2 * Log-Likelihood:	1347.2015

Test	Statistic	D.F.	P-Value
Score	54.9131	3	0.0000
Likelihood Ratio	62.6726	3	0.0000

Figure 7.156: Cox Proportional Hazards Statistical Results

Complex Sample Frequencies, Tables, and Means

The Frequencies (FREQ), Tables (TABLES), and Means (MEANS) commands in the Classic Analysis program perform statistical calculations that assume the data comes from simple random (or unbiased systematic) samples. More complicated sampling strategies are used in many survey applications. These may involve sampling features (i.e., stratification, cluster sampling, and the use of unequal sampling fractions). Surveys that include some form of complex sampling include the coverage surveys of the WHO Expanded Program on Immunization (EPI) (Lemeshow and Robinson, 1985) and CDC's Behavioral Risk Factor Surveillance System (Marks et al., 1985).

The CSAMPLE functions compute proportions or means with standard errors and confidence limits for studies where the data did not come from a simple random sample. If tables with two dimensions are requested, the odds ratio, risk ratio, and risk difference are also calculated.

Data from complex sample designs should be analyzed with methods that account for the sampling design. In the past, easy-to-use programs were not available for analysis of such data. CSAMPLE provides these facilities. It can form the basis of a complete survey system with an understanding of sampling design and analysis.

Complex Sample Frequencies

Dialog Box

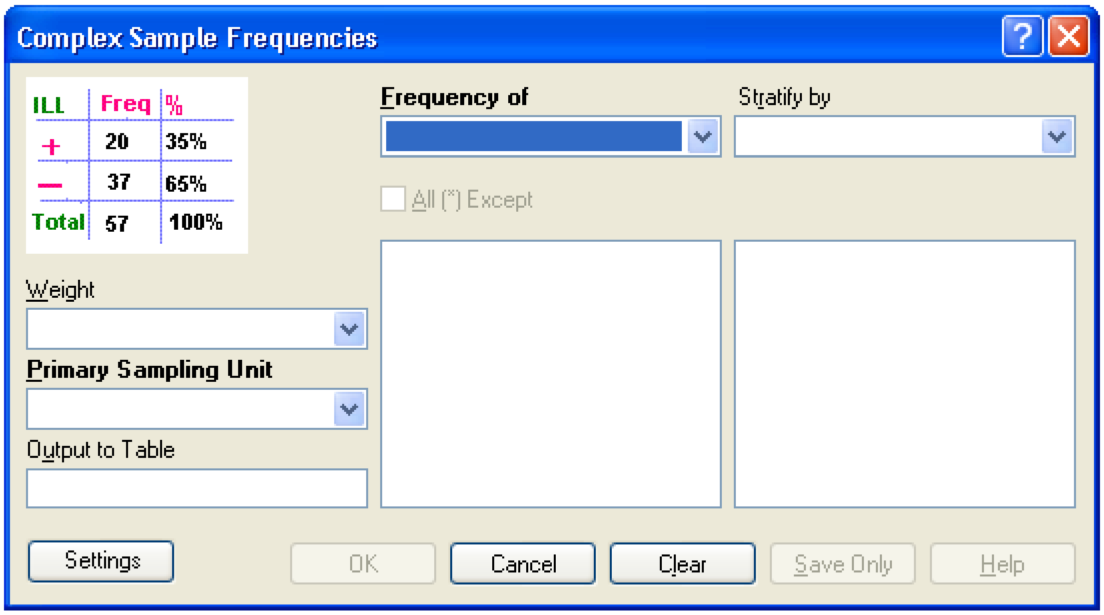


Figure 7.157: Complex Sample Frequencies Dialog Box

- A **Weight** variable is selected for use in weighted analyses.
- The **Output to Table** field identifies a data table to receive output from the command.
- **Frequency of** identifies the variable(s) whose frequency is computed.
- **All (*) Except** indicates that all the variables except those selected will have frequencies computed.
- **Stratify by** identifies the variable to be used to stratify or group the frequency data.
- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Save Only** saves the created code to the Program Editor but does not run the code.
- **Cancel** closes the dialog box without saving or executing a command.
- **Clear** empties the fields so information can be re-entered.
- **Help** opens the Help topic associated with the module being used. (Currently Disabled).

Complex Sample Means

Dialog Box

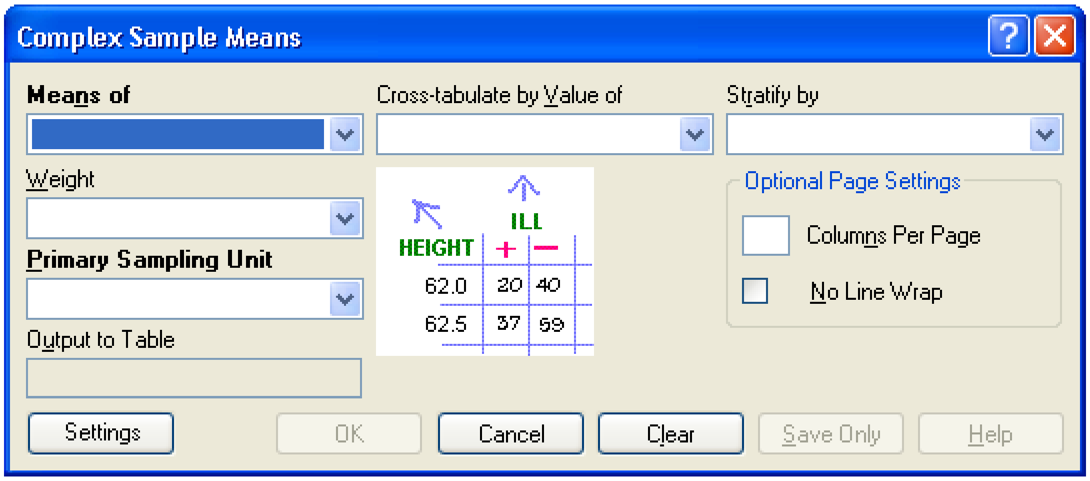


Figure 7.158: Complex Sample Means Dialog Box

- **Means of** identifies the variable whose mean is to be computed
- A **Weight** variable is selected for use in weighted analyses.
- The **Output to Table** field identifies a data table to receive output from the command. (Currently Disabled).
- **Cross-Tabulate by Value of** identifies the variable to be used to cross-tabulate the main variable.
- **Stratify by** identifies the variable to be used to stratify or group the frequency data.
- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Save Only** saves the created code to the Program Editor, but does not run the code.
- **Cancel** closes the dialog box without saving or executing a command.
- **Clear** empties the fields so information can be re-entered.
- **Help** opens the Help topic associated with the module being used. (Currently Disabled).

Complex Sample Tables

Dialog Box

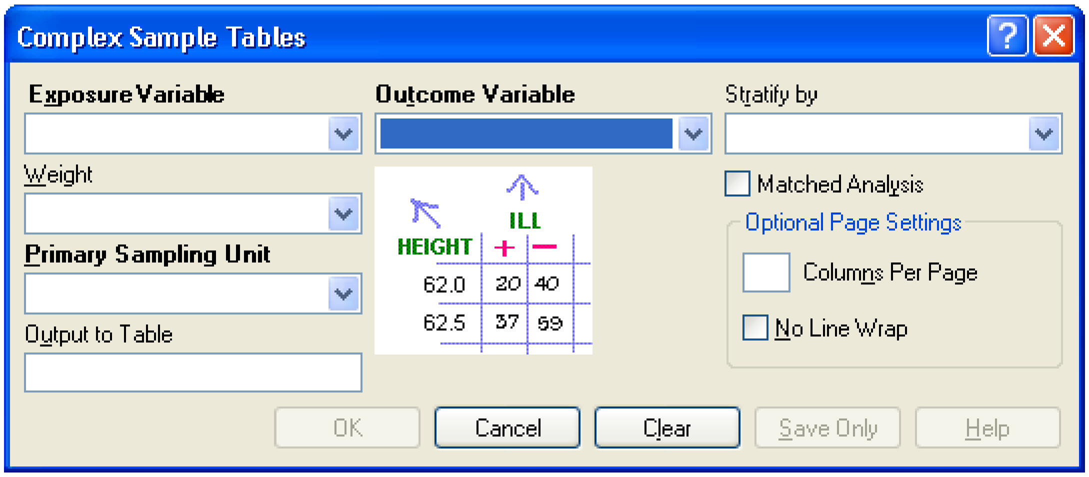


Figure 7.159: Complex Sample Tables Dialog Box

- **Exposure Variable** identifies the variable that will appear on the horizontal axis of the table. It is considered to be the risk factor (or * for all variables).
- A **Weight** variable is selected for use in weighted analyses.
- The **Output to Table** field identifies a data table to receive output from the command.
- **Outcome Variable** identifies the variable that will appear on the vertical axis of the table.
- **Stratify by** identifies the variable to be used to stratify or group the frequency data.
- **OK** accepts the current settings and data, and subsequently closes the form or window.
- **Save Only** saves the created code to the Program Editor, but does not run the code.
- **Cancel** closes the dialog box without saving or executing a command.
- **Clear** empties the fields so information can be re-entered.
- **Help** opens the Help topic associated with the module being used. (Currently Disabled).

THIS PAGE IS

INTENTIONALLY BLANK.

10. Maps: Visual Representation of Data by Location

Introduction

Epi Info™ 7 Maps is a versatile tool that geographically displays data on a map. The Maps tool has the ability to display multiple views from the same dataset. Datasets can be filtered or shown over a series of time using features in the Maps tool. Users can tailor these features to create a customized map containing public health data.

Information displayed in the main map window appears in the form of layers. Data layers exist in the form of case clusters, choropleth maps, or dot density maps. Reference layers add geographical boundaries and markers from shape files, a map server, or KML files. This format allows the Maps tool to uniquely identify and designate display settings from both internal and external data sources. Users may modify or filter map data using the data layers gadget located at the bottom of the main map window

Basic Tools

To open Maps, click **Create Maps** from the Epi Info™ 7 main menu, or select **Tools > Create Maps** from the main page navigation menu. Maps can also be accessed from the navigation menu at the top of the Enter screen. This section of the user guide provides instructions on how to use the Maps tool when accessed from the main menu. Accessing the Maps tool from the Enter navigation menu enables additional interactivity between the project data and the mapping tool. For additional information regarding the Maps tool's functionality from the Enter Data tool, refer to the Enter Data section of this user guide.

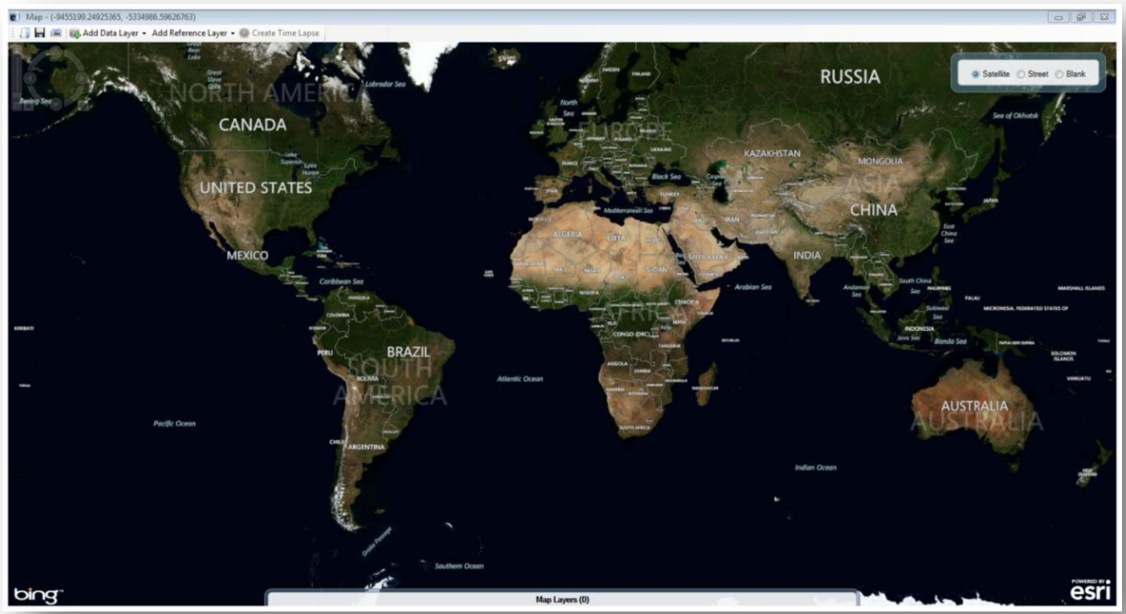


Figure 10.1: Main Map window

View Settings

The default setting in the main window is satellite view. The satellite view displays national boundaries, major roadways and geographic markings along with the region's topography. To change the view setting, click on the **Street** or **Blank** radio button at the top right corner of the screen. The street view displays similar geographic landmarks, roadways and boundaries without the region's topography. The blank view displays an empty canvas, which is beneficial when working with custom boundary files or adding reference layers (See Reference Layers).

Navigation Tools

To zoom in and out on a location, place the mouse over the desired location and rotate the mouse wheel forwards or backwards respectively. You can also zoom in or out by using the navigation panel at the top left corner of the screen. Click the **+** to zoom in or the – to zoom out. The navigation panel also allows you to click and drag the bar located in between the + and –to adjust the view. Directly to the right of the zoom option is a compass. Clicking on the arrows representing north, south, east and west will move the map location in the selected direction. You can also click and drag the map to adjust the screen. To rotate the view, click and drag the compass wheel until the map view meets your specifications.

Basic Functions

The map window contains the following tools to help create a meaningful representation of your data:

1. Add marker
2. Add zone
3. Add label
4. Remove all layers

Access each of these basic functions from the main map window.

Add Marker

A marker identifies a point of interest by placing a symbol on the map. Symbols are available in various shapes and sizes to identify multiple points of interest.

1. Right click on the **main map window** in the location where you would like to place the marker. A menu appears displaying the available options.
2. Select **Add marker.**

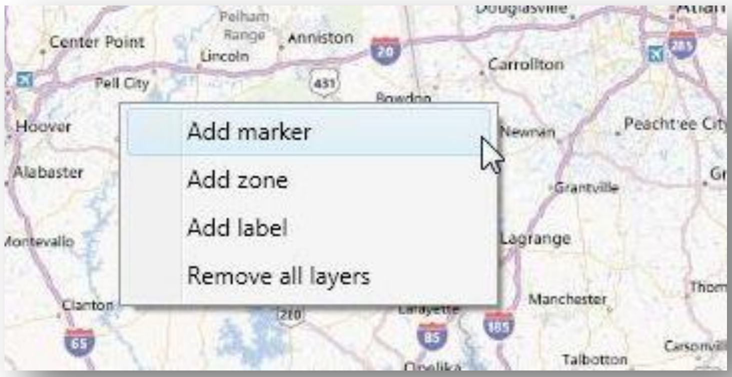


Figure 10.2: Basic Tools option

3. Select a **Style** from the Style drop-down list for the marker. The style will determine the marker's shape on the screen.

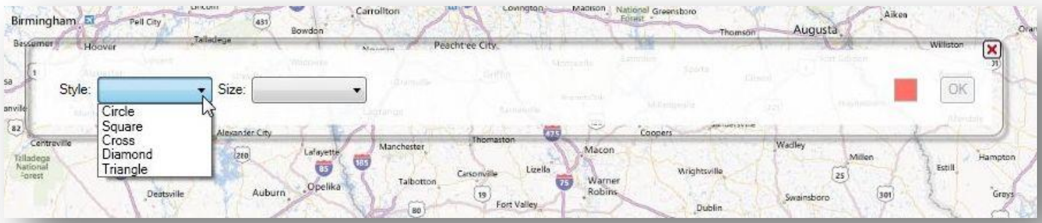


Figure 10.3: Marker styles

4. Select the **Size** of the marker from the Size drop-down list.

5. Select the **Color** of the marker by clicking on the red **Point Color** button to the right of the dialog box.

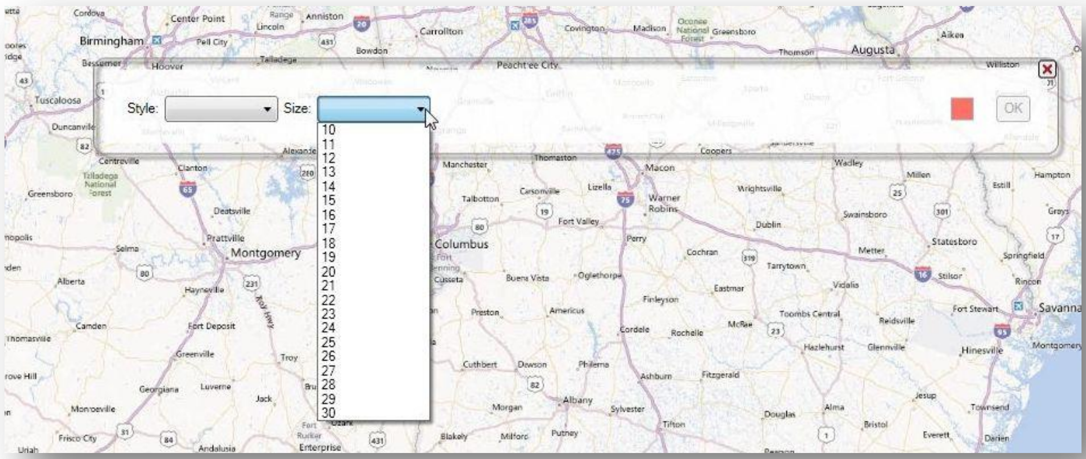


Figure 10.4: Marker size

6. Click **OK**.

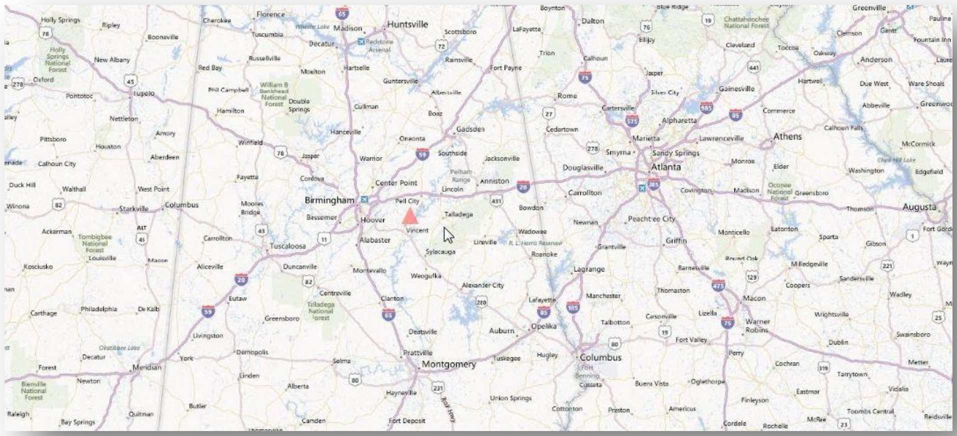


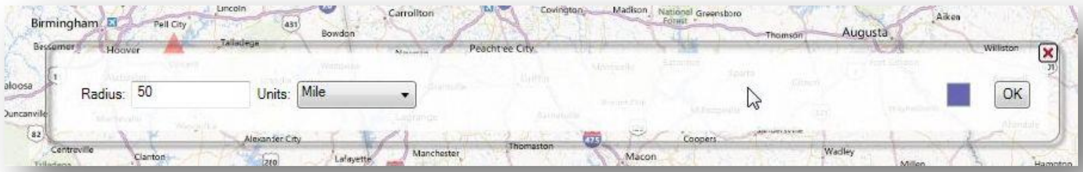
Figure 10.5: Marker displayed in Map window

Add Zone

A zone identifies a circular region on the map. This is useful when highlighting an area that centers on a point of interest or specifying an area with public health implications.

1. Right click on the main map window in the location where you would like to place the center of the zone.
2. Select **Add zone**.

3. Enter the desired **Radius**, **Units** and **Color**. Zones can be of varied size and color but are limited to a circular shape.



Radius: 50
Units: Mile

Figure 10.6: Zone settings

4. Click **OK**.

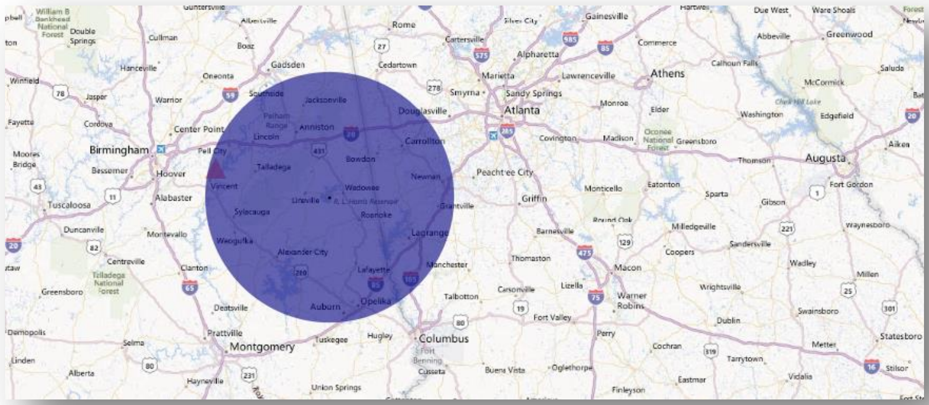


Figure 10.7: Zone displayed in map window

Note: Zones will appear distorted when placed closer to the north and south poles due to the area being re-projected from the globe onto a flat surface.

Add Label

1. Right click on the **main map window** in the location where you would like to place the label.
2. Select **Add label**.
3. Enter the label you would like to display in the textbox. Click the **ellipses** next to the **Font** textbox to select the desired **Font** and **Color**.

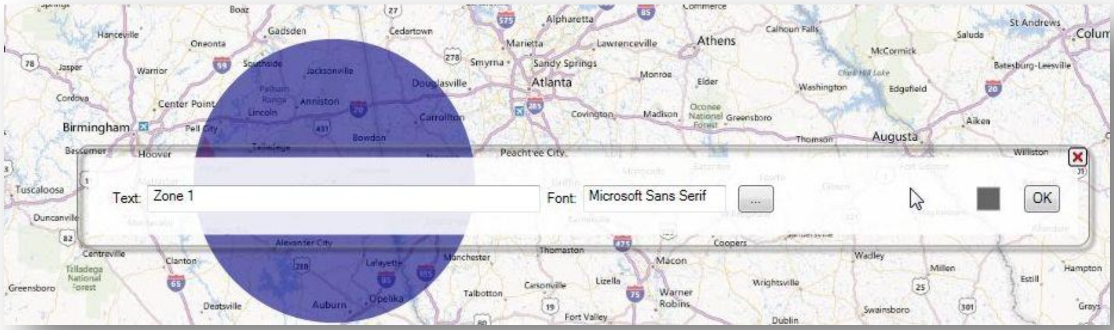


Figure 10.8: Label Settings

4. Click **OK**.

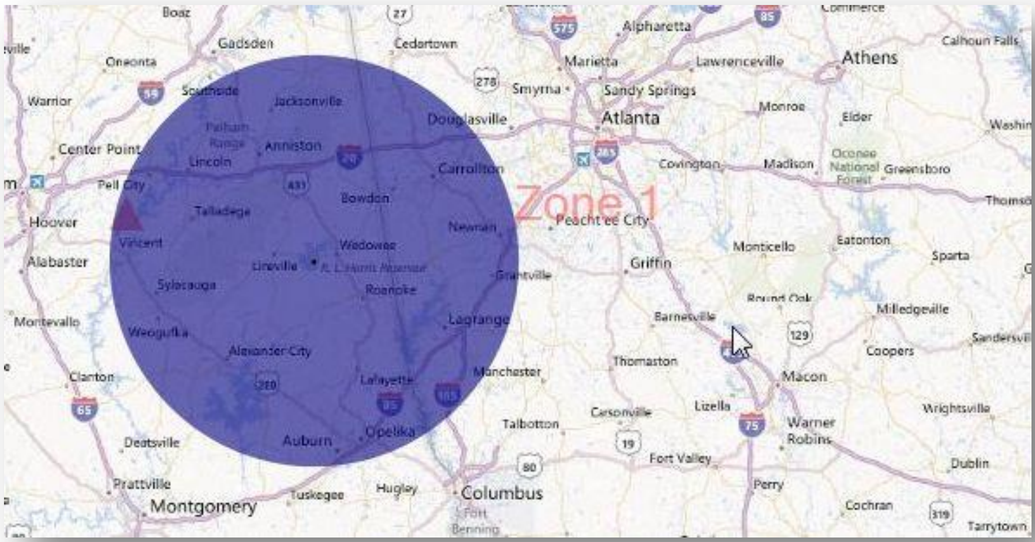


Figure 10.9: Label displayed in Map window

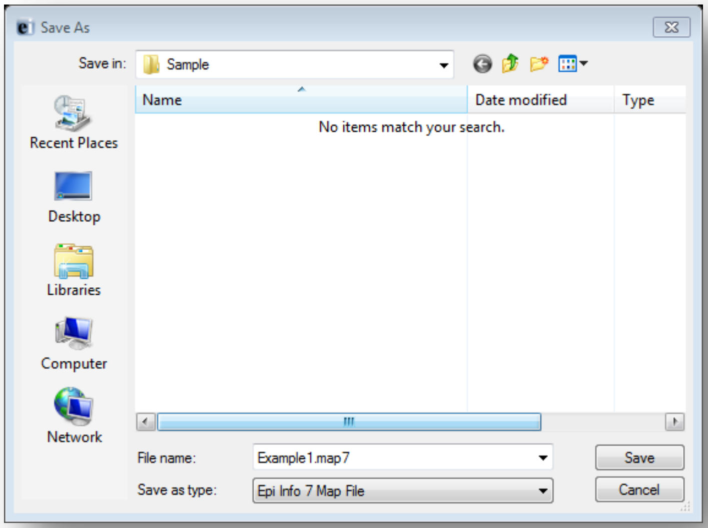
Remove All Layers

To clear the main map window of all layers, right click on the map and select **Remove all layers.**

Save Maps as Map File

Maps may be saved as either an image file (*.png) or an Epi Info™ 7 Map File (*.map7). The map file format saves the map along with the underlying data layers. The Epi Info™ 7 Map File can only be opened using the Epi Info™ 7 software. When opened, the map file will automatically update the map based on any changes to the dataset. Create and save the map file by clicking on the **Save** button in the Maps toolbar.

1. Click the **Save** button. A **Save As** dialog box opens.



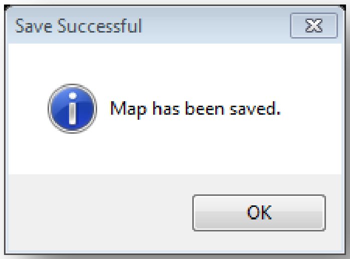
No items match your search.

File name: Example1.map7

Save as type: Epi Info 7 Map File

Figure 10.10: Save As dialog box

2. Create a **File name** and choose a file destination from the **Save in** drop-down list. The map is saved in the *.map7 file format, which can be opened later by Epi Info™.
3. Click the **Save** button. A **Save Successful** dialog box appears.



Map has been saved.

Figure 10.11: Save Successful

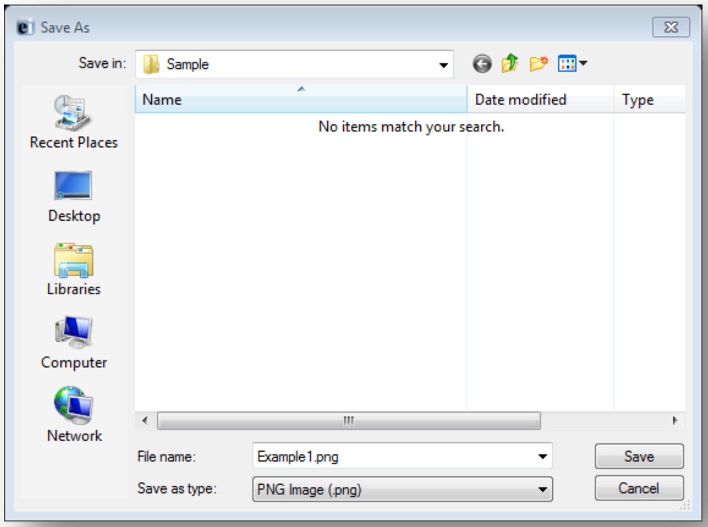
4. Click **OK** to exit.

Save Maps as an Image

The image format saves only the map image in a *.png file format. This image file can be inserted into documents or opened with a multitude of software applications. Save the map as an image by clicking on the **Save as Image** button in the Maps toolbar.

Note: The Epi Info™ software will not open or edit a map saved in image format.

1. Click the **Save as Image** button. A **Save As** dialog box appears.



No items match your search.

File name: Example1.png

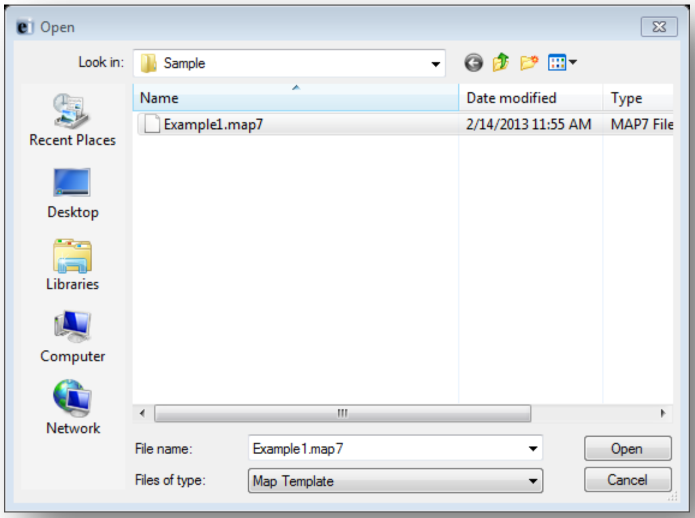
Save as type: PNG Image (.png)

Figure 10.12: Save As dialog box

2. Create a **File name** and choose a file destination from the **Save in** drop-down list. The map is saved in the *.png file format, which can be opened later with an image software application.
3. Click the **Save** button. A Save Successful dialog box appears.
4. Click **OK** to exit**.**

Open Maps

The Epi Info™ Maps tool will only open files in the Maps (*.map7) file format. To open a maps file, click the **Open** button in the Maps toolbar and select the desired file.



Name	Date modified	Type
Example1.map7	2/14/2013 11:55 AM	MAP7 File

File name: Example1.map7

Files of type: Map Template

Figure 10.13: Open Map file dialog box

Click **Open**. The map file updates and displays in the main map window.

Adding a Data Layer

Case Clusters
Case Clusters display case locations on a map based on geographic coordinates. Each dataset used to create a case cluster must contain numeric fields that Maps can designate as latitude and longitude. The software has the capability of geocoding a street address into these coordinates. In cases where geocoding is not applied, a street address alone will not be sufficient to create a case cluster. For additional information on geocoding, refer to the Form Designer section of this user guide.

In the main map window, large clusters appear as bigger circles with the total case count contained inside of them. Individual cases appear as single dots without a case count designation. The example below demonstrates how to create a case cluster map in street view with the data included in the E. coli project folder.

1. Select **Add Data Layer** from the navigation menu.
2. Select **Case Cluster** from the drop-down list.

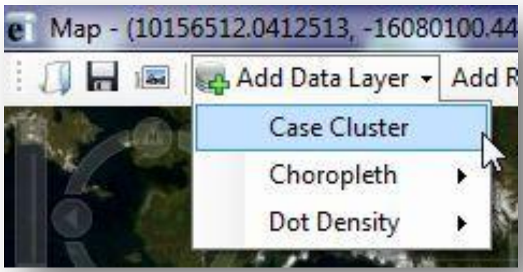


Figure 10.14: Case Cluster Data Layer

3. The **Data Source** dialog box appears.

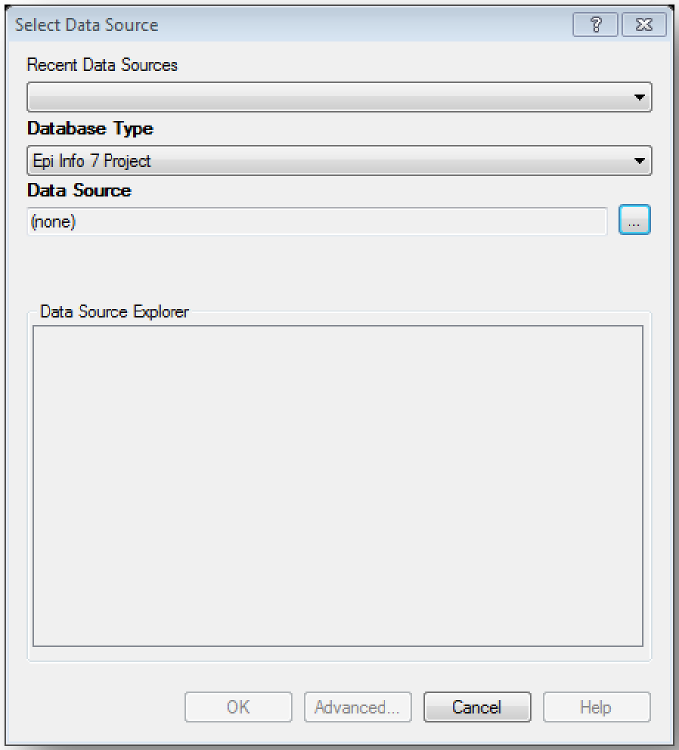


Figure 10.15: Select Data Source Dialog Box

4. Select the appropriate database from the **Database Type** drop-down list. There are multiple databases available from the drop-down list and it is imperative that the database type matches the file format that you are adding to Maps. For this demonstration, the default Epi Info 7 Project option is used.

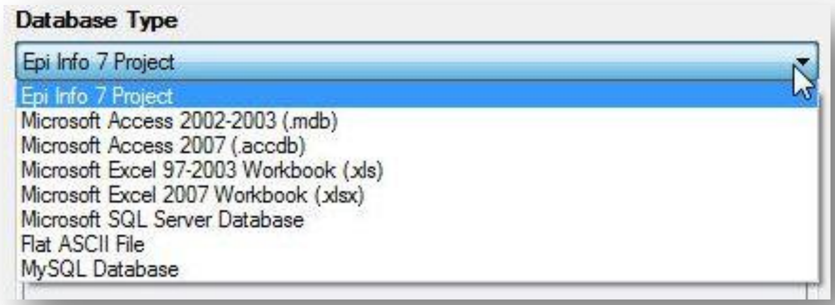
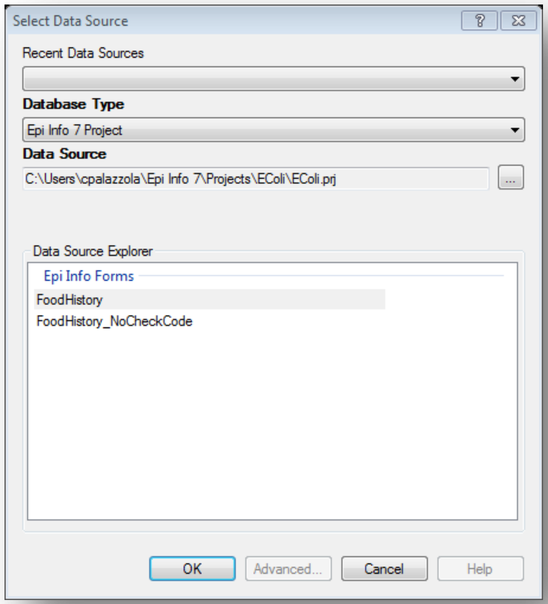

Database Type
Epi Info 7 Project
Epi Info 7 Project
Microsoft Access 2002-2003 (.mdb)
Microsoft Access 2007 (.accdb)
Microsoft Excel 97-2003 Workbook (.xls)
Microsoft Excel 2007 Workbook (.xlsx)
Microsoft SQL Server Database
Flat ASCII File
MySQL Database

Figure 10.16: Database Type

5. Click on the **ellipses** next to the Data Source field.
6. Select the project data that you would like to map. For this demonstration, select the **Ecoli project**.
7. Select the **Food History** form from the Data Source Explorer menu. Click **OK**.


Select Data Source

Recent Data Sources

Database Type: Epi Info 7 Project

Data Source: C:\Users\cpalazzola\Epi Info 7\Projects\EColi\EColi.prj

Data Source Explorer

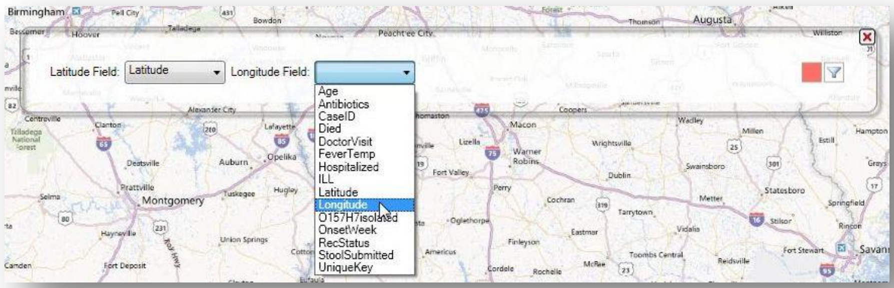
Epi Info Forms

FoodHistory

FoodHistory_NoCheckCode

Figure 10.17: Data Source Explorer dialog box

8. Select the field containing the Latitude coordinates (**Latitude)** from the Latitude drop-down list.
9. Select the field containing the Longitude coordinates (**Longitude)** from the Longitude drop-down list.


Latitude Field: Latitude

Longitude Field:

Figure 10.18: Latitude and Longitude drop-down list

10. The Maps tool will display all clusters based on latitude and longitude coordinates in red by default. You can change the color by clicking the red **Point Color** button and selecting a different color.

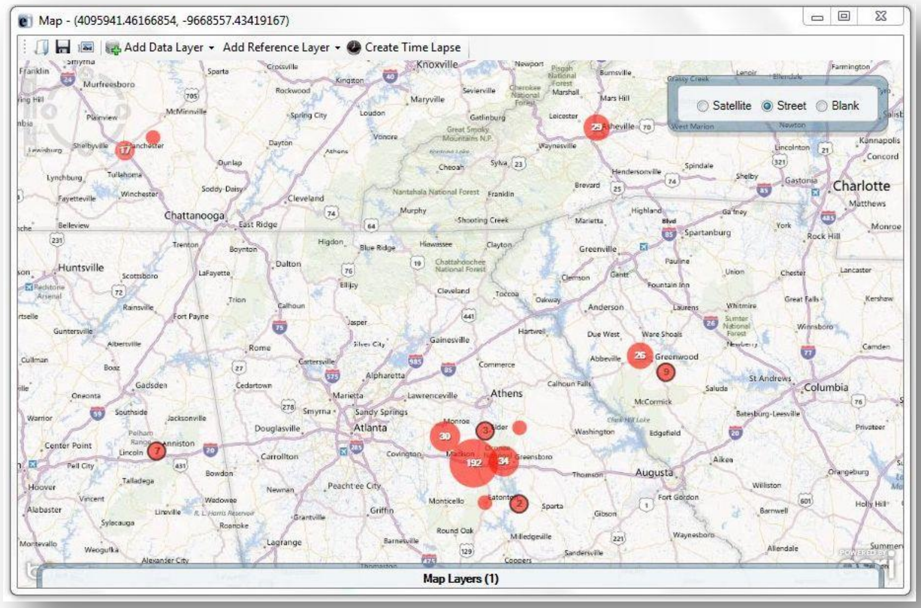


Figure 10.19: Case cluster map of E. coli cases

As the view zooms in, case clusters separate into smaller clusters and individual cases appear as single red dots.

Move the mouse over a small case cluster (less than 12 records) to flare the cluster outwards.

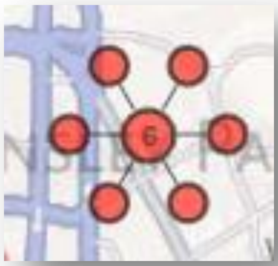


Figure 10.20: Expanded case cluster

The number six at the center of the cluster represents six different cases. When the mouse is over the case cluster, six smaller dots appear, each representing one of the six records inside of that cluster. This is useful if the same household contains multiple cases.

Data Filtering in Maps

Data filters in the Maps tool are used to select a subset of data by specifying and applying certain conditions. This allows the user to show the effect of a variable on the geographic distribution of cases. To access the data filter tool, move your cursor over the **Map Layers** gadget at the bottom of the main map window. Currently there is one case cluster layer in red with no filters.

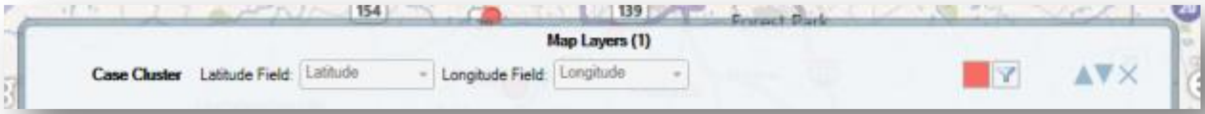


Figure 10.21: Map Layers gadget

The following example demonstrates how to apply a filter to the **Age** variable from the data layer added in the previous example.

1. Click the **Data Filters** button. The Data Filters gadget appears.

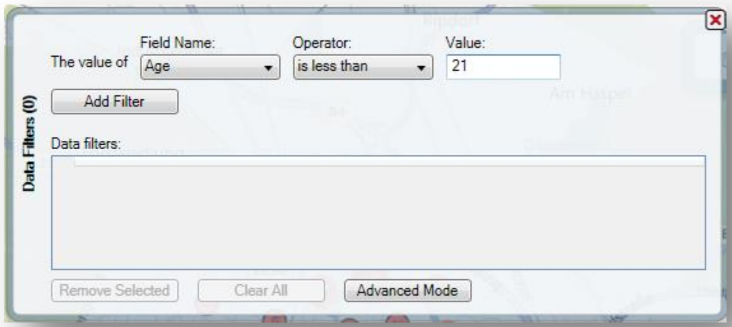
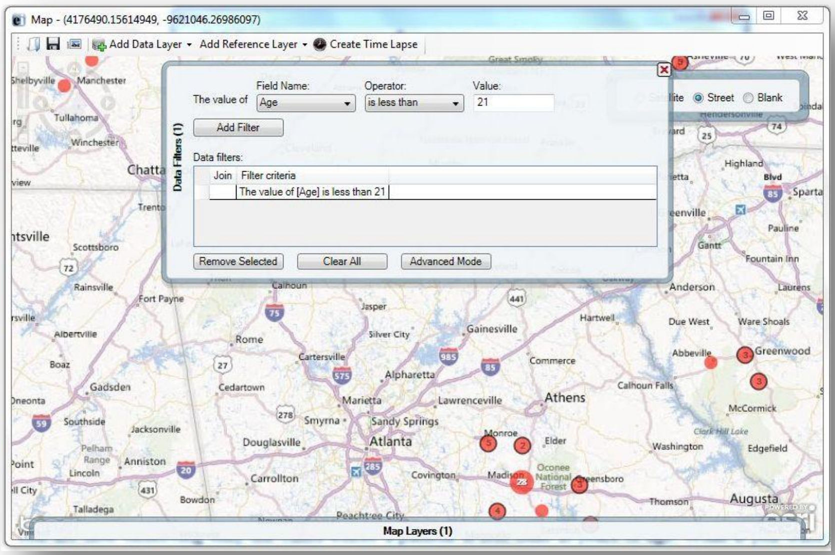


Figure 10.22: Data Filters dialog box

2. From the **Field Name** drop-down list, select **Age**.
3. From the **Operator** drop-down list, select **Is Less Than**.
4. Enter **21** into the **Value** textbox.
5. Click the **Add Filter** button. The **Data Filter** gadget displays one row in the Data filters table.



The value of Field Name: Age Operator: is less than Value: 21

Data filters:

Join	Filter criteria
	The value of [Age] is less than 21

Figure 10.23: Map window with Data Filter

6. To close the gadget, click the **X** button. The map displays only cases where the patient's age is less than 21.

Stratifying Data

Filters are a powerful tool but in certain instances, it is best to display the full dataset in groupings. This requires stratifying the data. After completing the previous steps, the map displays the patient population that is less than 21 years of age. To create the stratified map, another layer is required to show an additional age group in the patient population. In the following example, a filter is applied to an additional layer based on the patient's age. Blue clusters will represent the records that meet the specified condition.

1. From the toolbar at the top of the map window, select **Add Data Layer > Case Cluster**.
2. Select the **Food History** form in the E. coli project. A small gadget appears with Latitude and Longitude drop-down lists.
3. Before selecting Latitude and Longitude, click the red **Point Color** button to change the layer's color to blue.

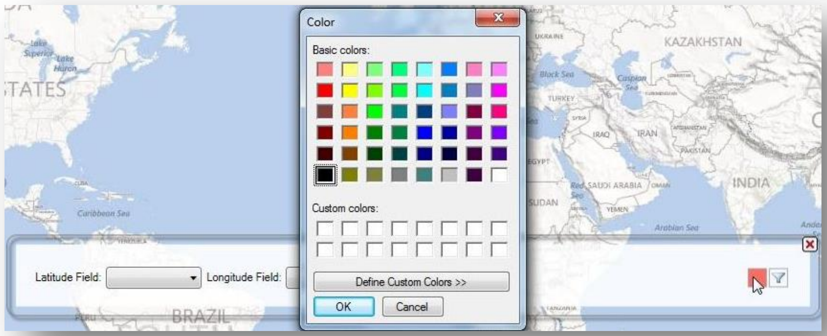
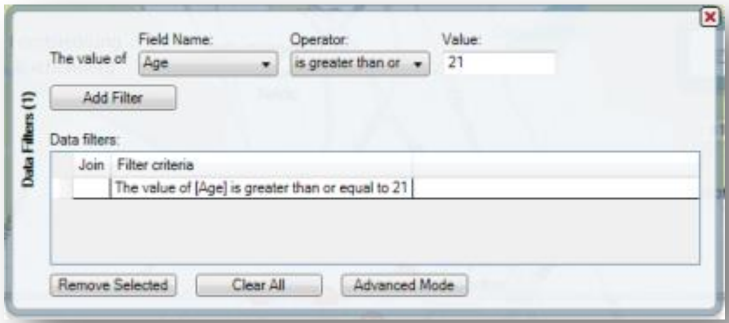


Figure 10.24: Point Color Menu

4. Click the **Data Filters** button. The Data Filters gadget appears. From the **Field Name** drop-down list, select **Age**.
5. From the **Operator** drop-down list, select **Is Greater Than or Equal To**.
6. Enter **21** into the **Value** textbox.
7. Click the **Add Filter** button.


Data Filters (1)

The value of — Field Name: Age — Operator: is greater than or — Value: 21

Add Filter

Data filters:

Join	Filter criteria
	The value of [Age] is greater than or equal to 21

Figure 10.25: Stratified Data Filters gadget

8. To close the Data Filters gadget, click the **X** in the top-right corner.
9. From the **Latitude** drop-down list, select **Latitude**.
10. From the **Longitude** drop-down list, select **Longitude**. The stratified map displays both age groups by color.

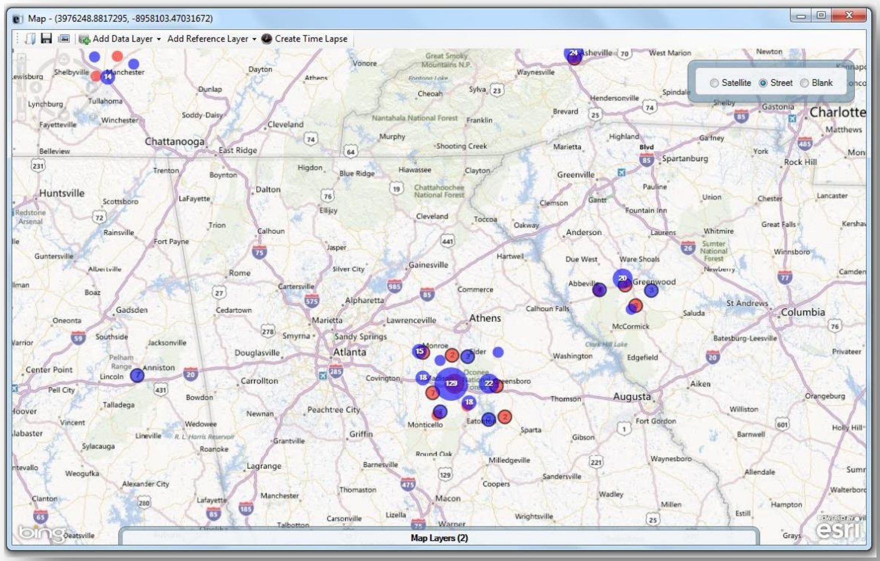


Figure 10.26: Stratified Map window

The blue clusters represent cases where the patient is 21 years of age or older, while the red clusters represent cases where the patient is less than 21. The Maps tool allows for multiple age groupings and can also stratify the dataset based on other variables (e.g., whether the patient is male or female, foods eaten, etc.).

Note that the Map Layers gadget at the bottom of the screen displays a (2) instead of a (1). This indicates that there are two different layers with corresponding filters.

Time Lapse

When a case cluster layer is added to Maps, the tool displays all cases contained in the dataset (If a filter is applied, Maps will display all cases that meet the selected filter criteria). The Time Lapse tool creates a dynamic environment to show how the dataset transforms over time. To enable this feature, load a data layer with a time variable.

1. Add a **Case Cluster** data layer (See Adding a Data Layer). Use the Food History form in the E. coli project for the following example.
2. Click **Create Time Lapse.**

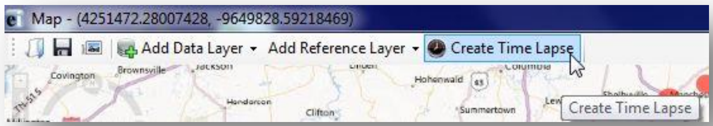


Figure 10.27: Create Time Lapse

3. Select the **OnsetDate** from the **Time Variable** from the drop-down list. Click **OK**.

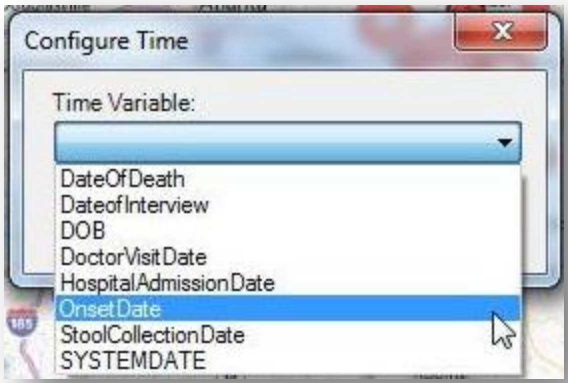


Figure 10.28: Configure Time dialog box

4. To show the progression of the symptom onset date over time, click the **Run** button to the left of the timeline.

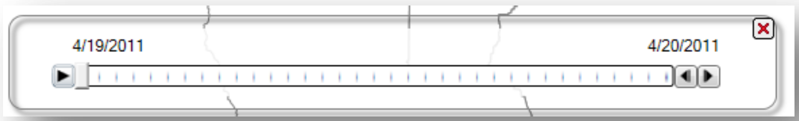


Figure 10.29: Timeline

5. The main map window clears and begins adding the cumulative number of cases in accordance with the timeline. The figure below displays the number of cases in the dataset from 4/19/2011 to 5/17/2011.

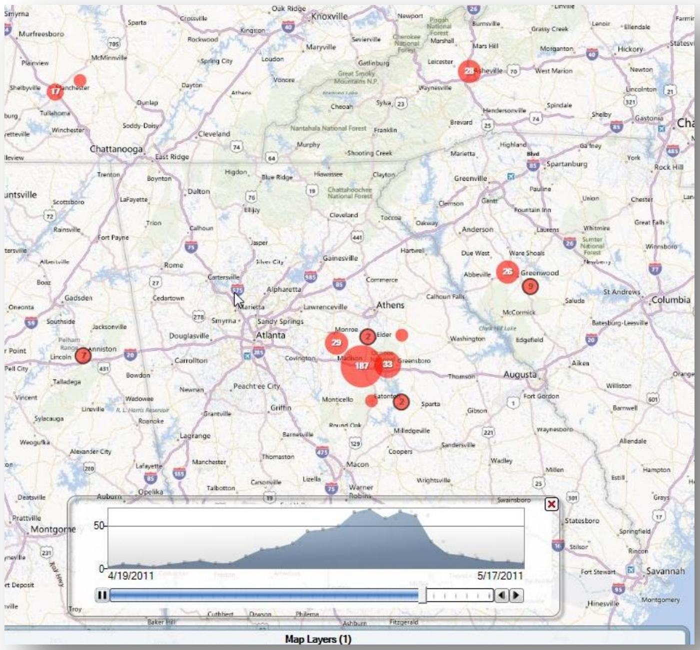


Figure 10.30: Time Lapse display

6. It is important to note that the distribution of cases displays at the bottom of the screen. At any time, press pause to show the cumulative dataset at that time interval. The buttons to the right of the timeline move the time series forwards and backwards.

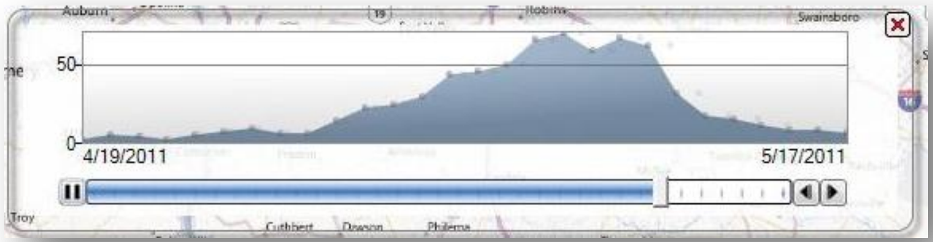


Figure 10.31: Time Series Distribution

Creating Choropleth and Dot Density Maps with Boundaries

Maps can create a choropleth or dot density map by combining a dataset with three boundary formats: shape file, map server, or KML (Keyhole Markup Language). See below for map/boundary format compatibility:

- Choropleth: Shape file
- Choropleth: Map server
- Choropleth: KML

- Dot density: Shape file
- Dot density: Map server
- Dot density: KML

The boundaries are independent of the dataset but are joined with database keys.

Note: The dataset and boundary file must contain the proper database keys and the user must designate the proper key from each drop-down list to create a choropleth or dot density map.

Key descriptions are as follows:

- **Feature Key** – designates the variable in the boundary set that will match a corresponding variable in the dataset.
- **Data Key** – designates the variable in the dataset that will match a corresponding variable in the boundary set.
- **Value Field** – designates the value being displayed.

Shape Files

A shape file stores non-topological geometric and spatial information in vector format. These files are simple to use but lack complex data elements. Various sources attach additional information or tables to shape files for more advanced analysis.

KML

KML is an open source specification for describing geographic data. Like shape files, KML files contain instructions used by mapping tools to draw boundaries, points, and other feature sets. A benefit to using KML files is that they can be edited using simple text editors.

Map Servers

Map server is a platform used to publish spatial and geographic data to the internet. One advantage of the map server format is the capability of creating a central repository for mapping data with relational database management systems.

Choropleth Map using Map Server Boundaries

A choropleth map uses graded differences in shading or color to display variations of a variable across a geographic area. The color gradient typically spans from one color to another or from light to dark.

The example below demonstrates how to combine a map server feature set for the state of New York with the Lyme data set included with Epi Info to create a choropleth map.

Note: This example uses case based data, where each row in the data set represents an individual case, rather than aggregate data.

1. Select **Add Data Layer** from the navigation menu.
2. Select **Choropleth>With Map Server Boundaries** from the drop-down list.

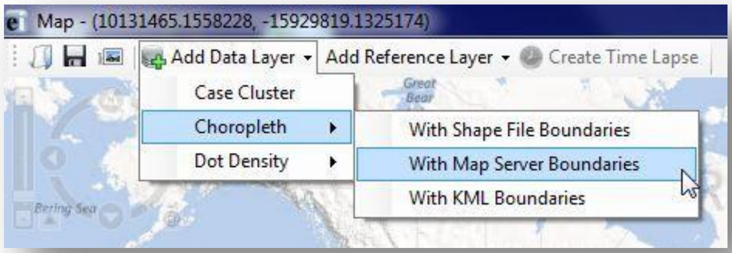
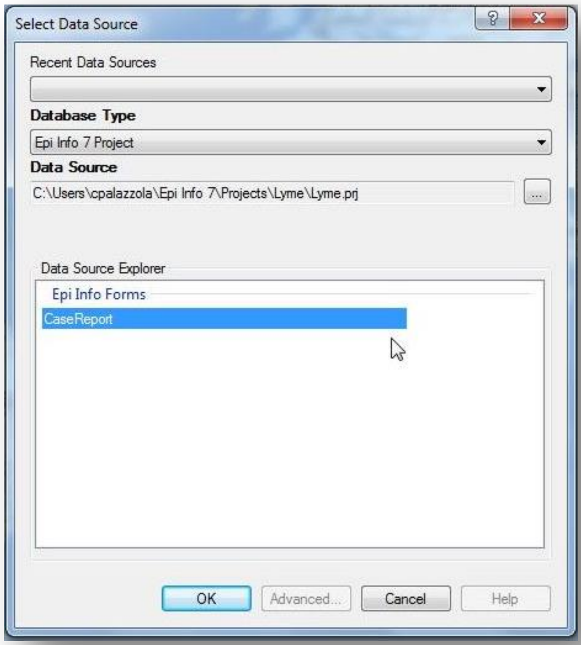


Figure 10.32: Choropleth using Map Server Boundaries

3. The Data Source dialog box appears. Select **Database Type** from the drop-down list.
4. Click on the **ellipses** next to the Data Source field. Select the Lyme project from the Epi Info™ 7 Projects folder.
5. Select the **Case Report** form. Click **OK**.



Database Type
Epi Info 7 Project
Data Source
C:\Users\cpalazzola\Epi Info 7\Projects\Lyme\Lyme.prj
Data Source Explorer
Epi Info Forms
CaseReport

Figure 10.33: Choropleth Data Source dialog box

6. Click on the **Browse Map Server** button.

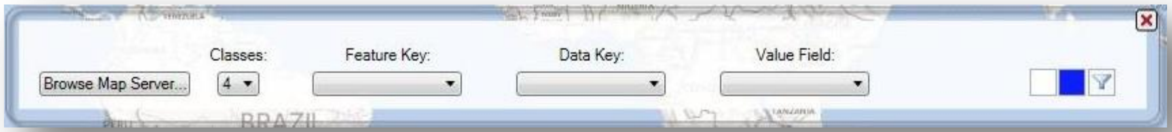


Figure 10.34: Browse Map Server

7. Select **NationalMap.gov – New York County Boundaries** from the drop-down list. Click **OK**.

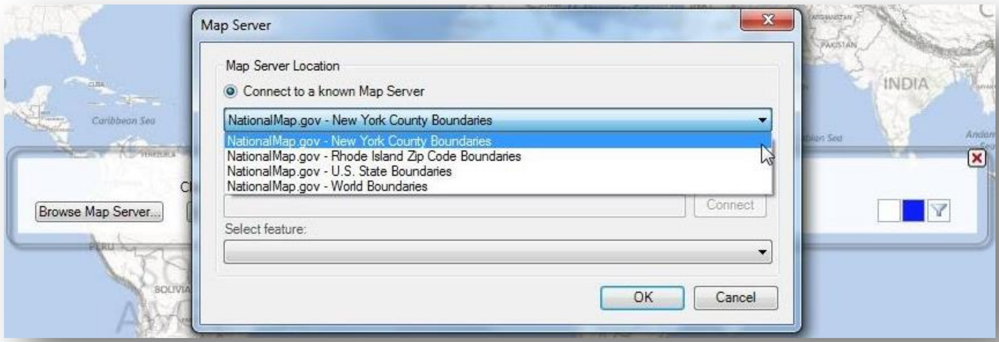

Map Server
Map Server Location
Connect to a known Map Server
NationalMap.gov - New York County Boundaries
NationalMap.gov - New York County Boundaries
NationalMap.gov - Rhode Island Zip Code Boundaries
NationalMap.gov - U.S. State Boundaries
NationalMap.gov - World Boundaries
Connect
Select feature:
OK
Cancel
Browse Map Server...

Figure 10.35: Map Server Location

8. The New York county boundaries appear on the map.
9. Select the number of **Classes** you want to display. By clicking on **four** from the drop-down list, the choropleth will designate values into quartiles.

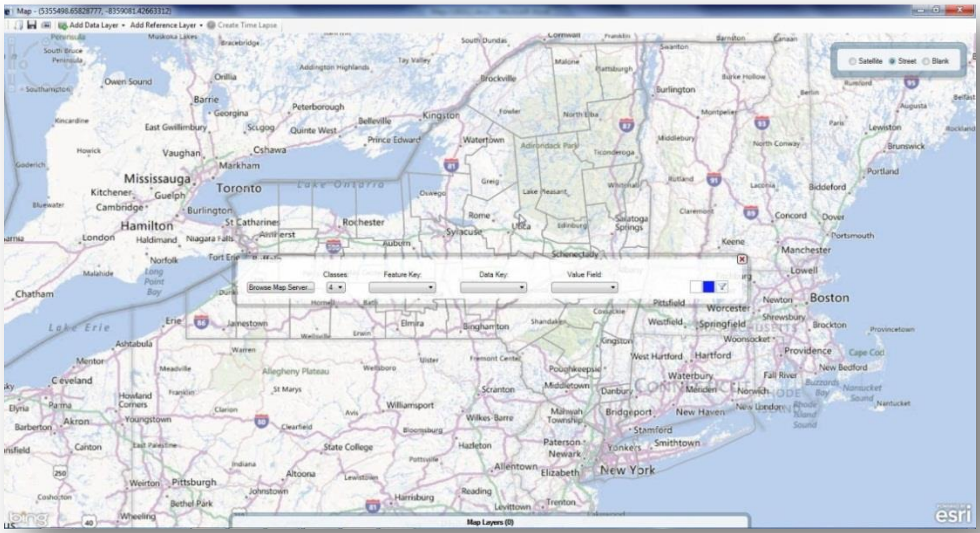


Figure 10.36: Map Server Boundaries

10. Select **COUNTY_NAME** from the **Feature Key** drop-down list. The feature key connects County_Names in the map server file to the dataset.

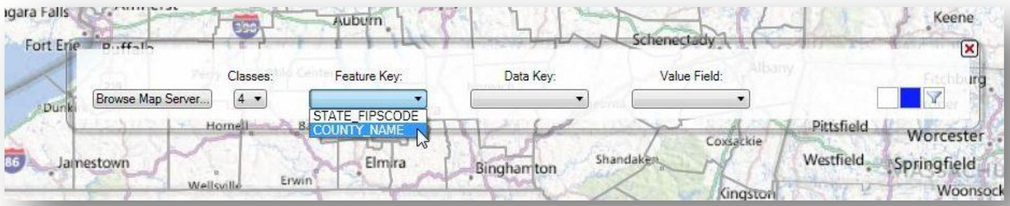


Figure 10.37: Map Server Feature Key

11. Select **County** from the **Data Key** drop-down list. The data key connects the dataset field, County, to the map server feature key COUNTY_NAME.

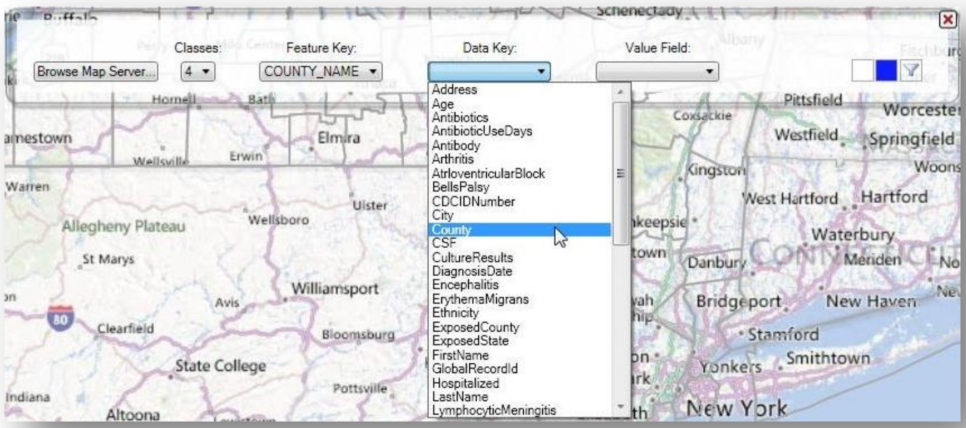


Figure 10.38: Map Server Data Key

12. Select <**Record Count>** from the **Value Field** drop-down list. Record Count is a system variable that provides the totals for the number of records in each boundary. This action will populate the choropleth based on the record count value.

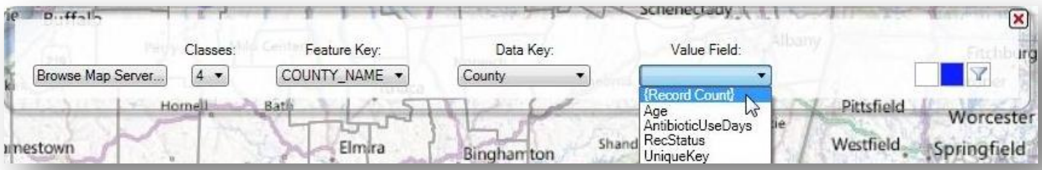


Figure 10.39: Map Server Value Field

The choropleth map appears showing the number of records in each New York County. Counties with the highest tier of records are dark blue while Counties containing the least amount of records are white.

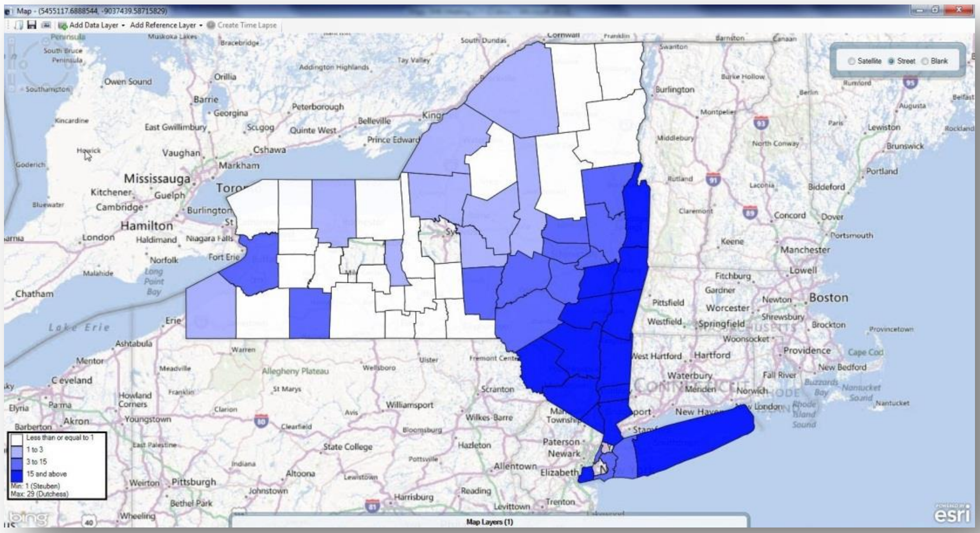


Figure 10.40: Choropleth in Street view

13. Select **Blank** view to display the choropleth with a white background.

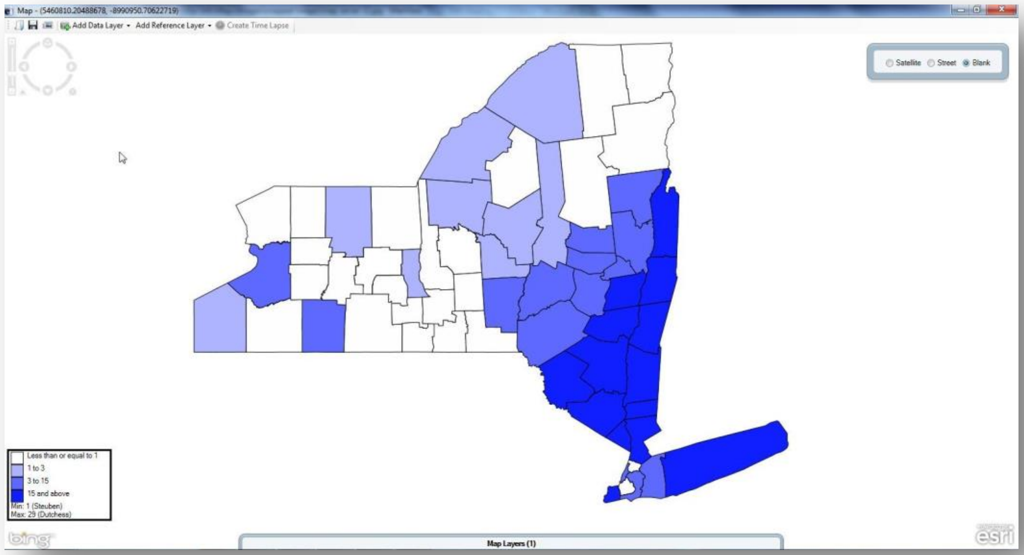


Figure 10.41: Choropleth in Blank view

14. To edit the color scheme or to add a filter, place the mouse over the Map Layers gadget located at the bottom of the screen.

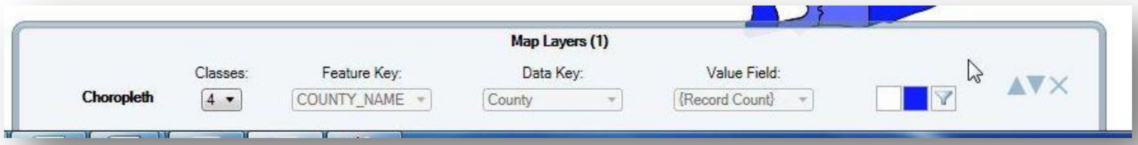


Figure 10.42: Map Layers Tab

Choropleth Map using KML Boundaries

The following demonstration uses aggregate data extracted from the 2011 U.S. Census' American Community Survey to show the percentage of the population lacking health insurance in counties in Maryland.

Note: This example uses aggregate data, which lists each geographic area once and contains a column for the aggregate value.

1. Select **Add Data Layer** from the navigation menu.
2. Select **Choropleth>With KML Boundaries** from the drop-down list.

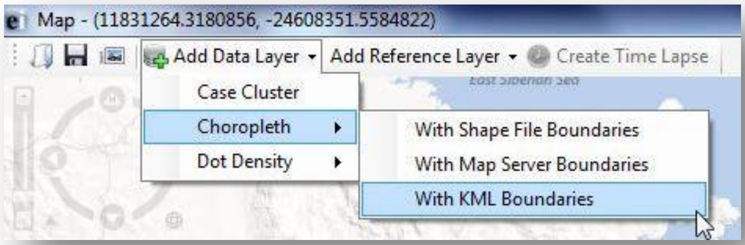


Figure 10.43: Choropleth using KML Boundaries

3. The **Data Source** dialog box appears. Select **Microsoft Excel 97-2003 Workbook** from the **Database Type** drop-down list.

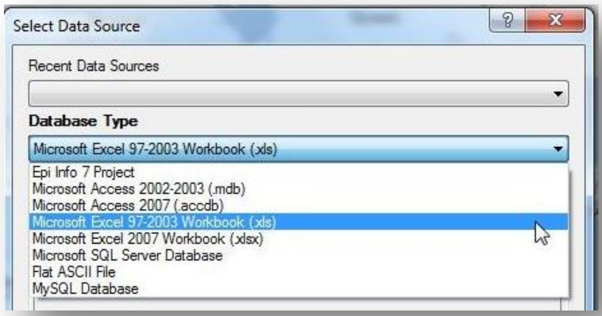
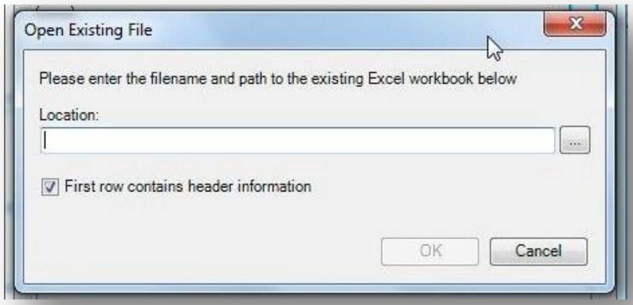

Select Data Source
Recent Data Sources
Database Type
Microsoft Excel 97-2003 Workbook (xls)
Epi Info 7 Project
Microsoft Access 2002-2003 (.mdb)
Microsoft Access 2007 (.accdb)
Microsoft Excel 97-2003 Workbook (xls)
Microsoft Excel 2007 Workbook (xlsx)
Microsoft SQL Server Database
Flat ASCII File
MySQL Database

Figure 10.44: Database Type drop-down list

4. Click the **ellipses** next to the Data Source field. The **Open Existing File** dialog box appears.
5. Click the **ellipses** next to the **Location** textbox to browse for the KML file you would like to map. Select the **KML_Example** project.


Open Existing File

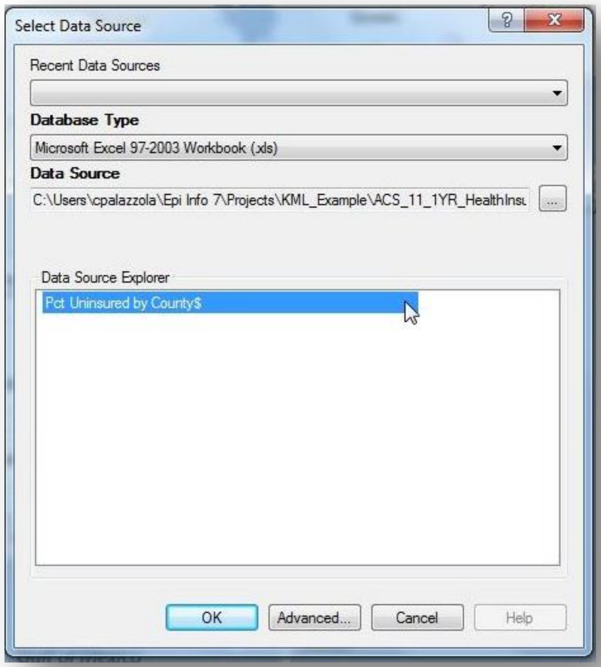
Please enter the filename and path to the existing Excel workbook below

Location:

☑ First row contains header information

Figure 10.45: Open Existing File dialog box

6. Select **ACS_11_1YR_HealthInsurance**. Click **Open**.
7. Select the **Pct Uninsured by County$** Form. Click **OK**.


Select Data Source

Recent Data Sources

Database Type

Microsoft Excel 97-2003 Workbook (xls)

Data Source

C:\Users\cpalazzola\Epi Info 7\Projects\KML_Example\ACS_11_1YR_HealthInsu

Data Source Explorer

Pct Uninsured by County$

Figure 10.46: Choropleth Data Source dialog box

8. Click the **Specify KML** button.

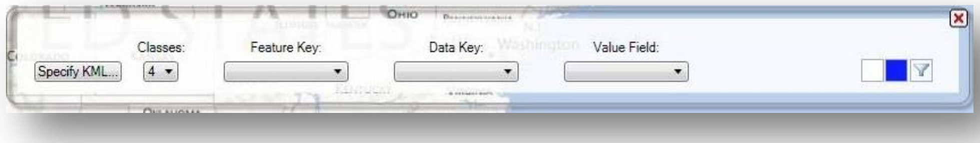
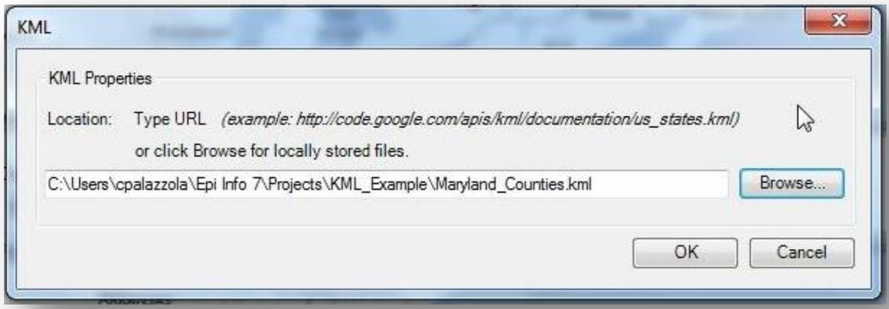


Figure 10.47: Specify KML

9. Click the **Browse** button and select the **Maryland_Counties.kml** file under the KML_Example project folder. Click **OK**.



Location: Type URL *(example: http://code.google.com/apis/kml/documentation/us_states.kml)*
or click Browse for locally stored files.

C:\Users\cpalazzola\Epi Info 7\Projects\KML_Example\Maryland_Counties.kml

Figure 10.48: KML Location

10. A blank Maryland state map appears.
11. Select the number of **Classes** you want to display. By clicking on **four** from the drop-down list, the choropleth will designate values into quartiles.

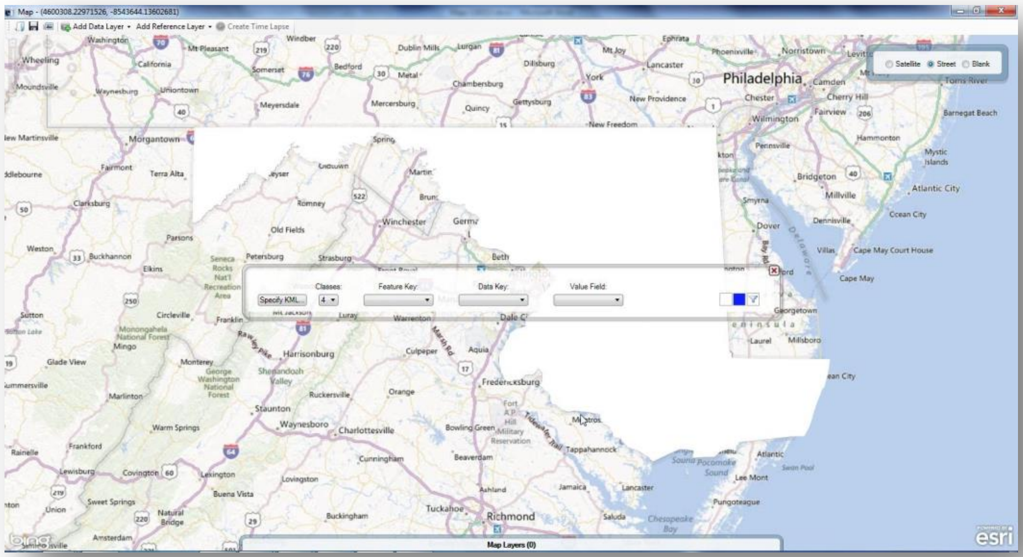


Figure 10.49: KML Boundaries

12. Select **County_Code** from the **Feature Key** drop-down list. The feature key connects County_Code in the KML file to the dataset.

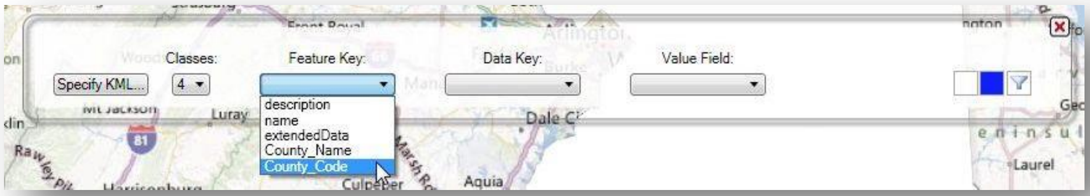


Figure 10.50: KML Feature Key

13. Select **Geo#id2** from the **Data Key** drop-down list. The data key connects the dataset field, Geo#id2, to the KML feature key County_Code.

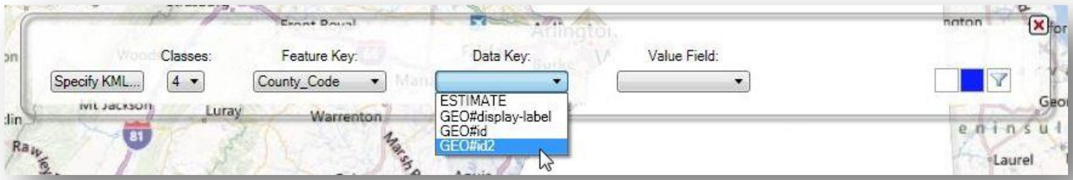


Figure 10.51: KML Data Key

14. Select **Estimate** from the **Value Field** drop-down list. This action will populate the choropleth based on the Estimate value.

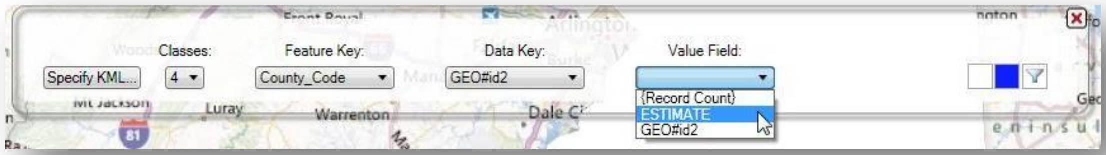


Figure 10.52: KML Value Field

The choropleth map appears showing the estimated percentage of the population lacking health insurance in each Maryland County.

Note: Territories that do not contain any data will remain uncolored.

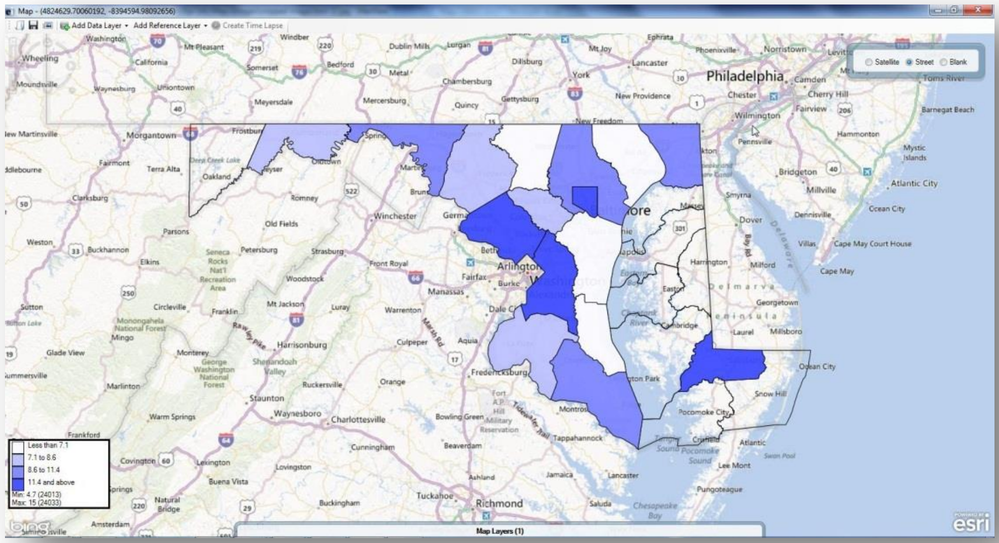


Figure 10.53: Choropleth in Street view

15. Select **Blank** view to display the choropleth with a white background.

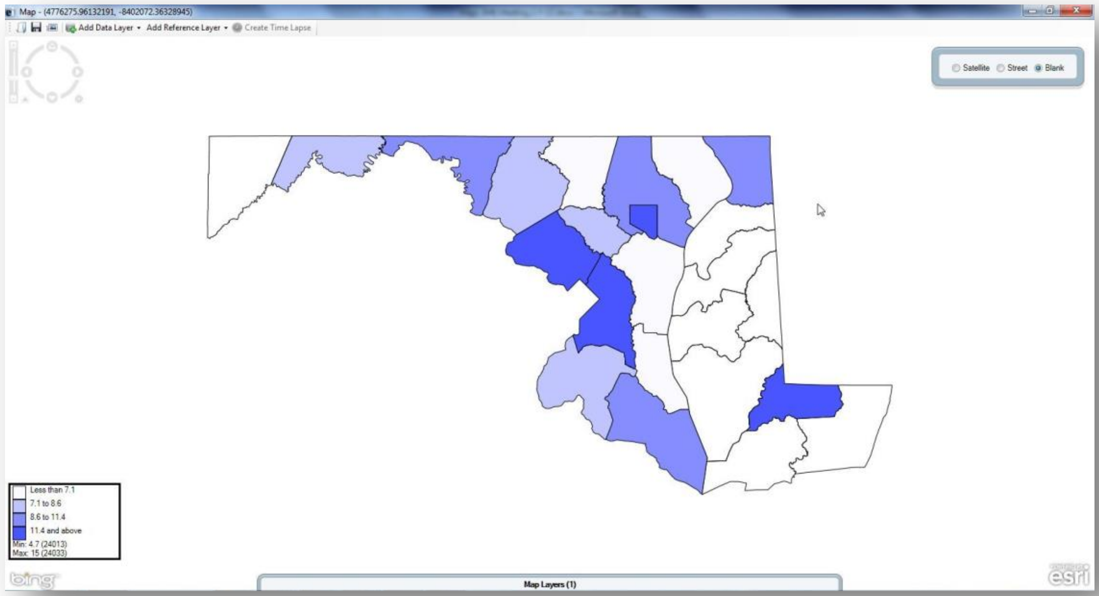


Figure 10.54: Choropleth in Blank view

16. To edit the color scheme or to add a filter, place the mouse over the Map Layers gadget located at the bottom of the screen.

Dot Density Map using Shape File Boundaries

To display densities across geographic boundaries, select **Dot Density** from the data layers drop-down list. Maps will populate the dot density map according to the dot value selected in step eight. Increasing the dot value increases the value each dot represents. The dot density map populates each dot randomly within a set of boundaries to display the density. The example below demonstrates how to create a dot density map with data from a Vital Statistics report published by the Mexico Ministry of Health using a shape file.

1. Select **Add Data Layer** from the navigation menu.
2. Select **Dot Density>With Shape File Boundaries** from the drop-down list.

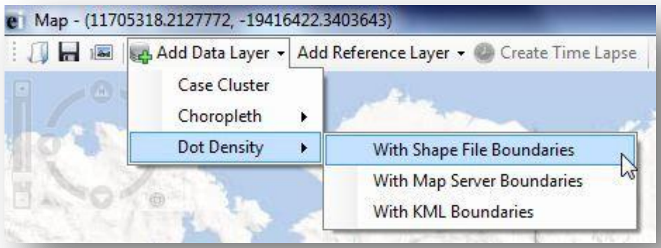


Figure 10.55: Dot Density with Shape File Boundaries

3. The **Data Source** dialog box appears. Select **Database Type** from the drop-down list.
4. Click on the **ellipses** next to the Data Source field. Select the project data that you would like to map. The following demonstration uses the Sample project.
5. Select the **MexMap95** form. Click **OK**.

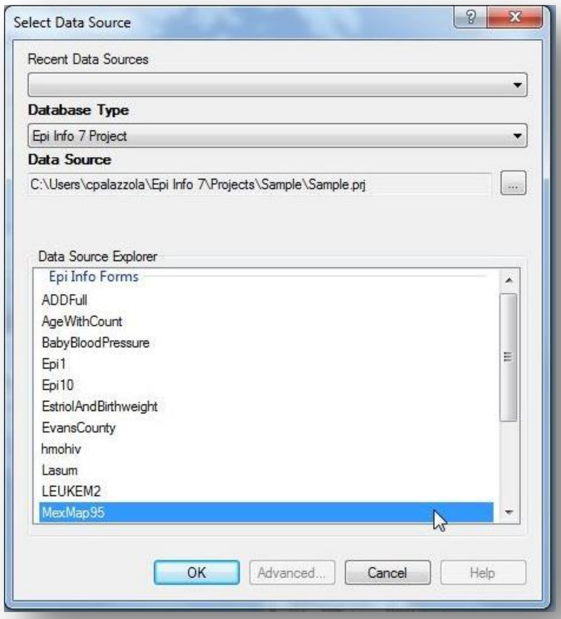


Figure 10.56: Shape File Data Source dialog box

6. Click the **Browse Shape File** button.

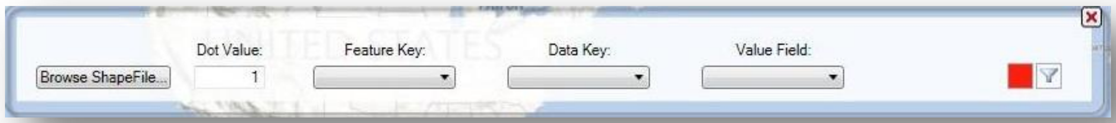


Figure 10.57: Browse Shape File

7. Select **MXState.shp** under the Sample project folder. The MexMap95 shape file boundaries appear on the map.
8. Enter the amount, as an integer, that each dot should count for under **Dot Value**. In this example, the dots are associated with the percentage of teen pregnancy in each state. By specifying “1”, each dot will represent a single teen birth percentage point.

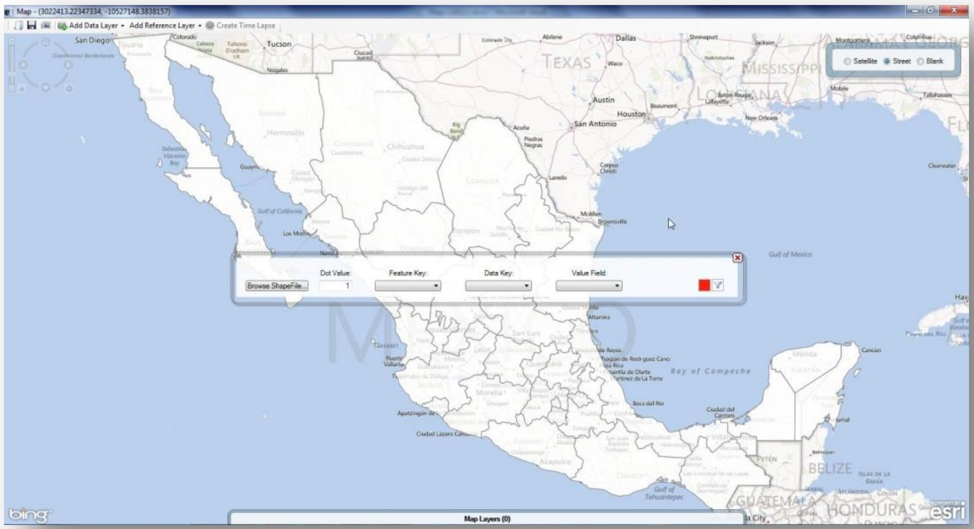


Figure 10.58: Shape File boundaries

9. Select **name** from the **Feature Key** drop-down list. The feature key connects name in the shape file to the dataset.

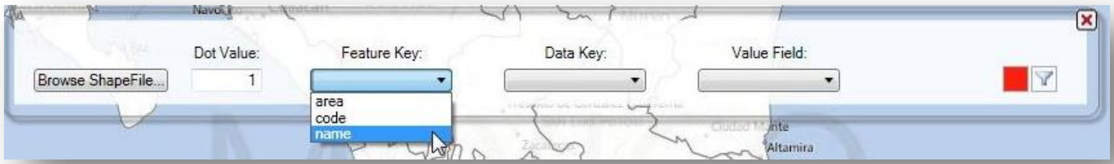


Figure 10.59: Shape File Feature Key

10. Select **STATE** from the **Data Key** drop-down list. The data key connects the dataset field, STATE, to the shape file feature key name.

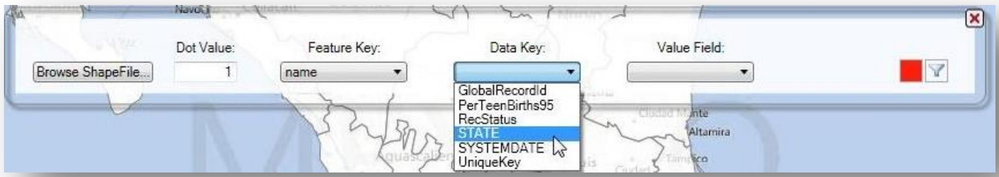


Figure 10.60: Shape File Data Key

11. Select **PerTeenBirths95** from the **Value Field** drop-down list. This populates the dot density map based on the percentage of teen births in that state.

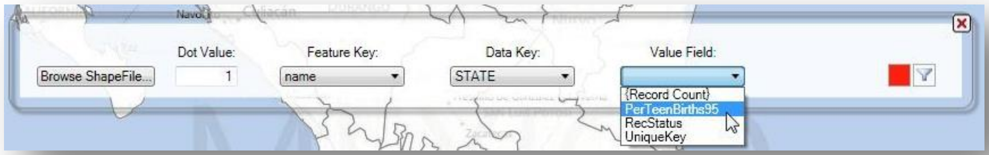


Figure 10.61: Shape File Value Field

Since one was selected as the dot value, each red dot that appears represents a single percentage point. Areas with a higher concentration of red dots represent a higher percentage of teen births.

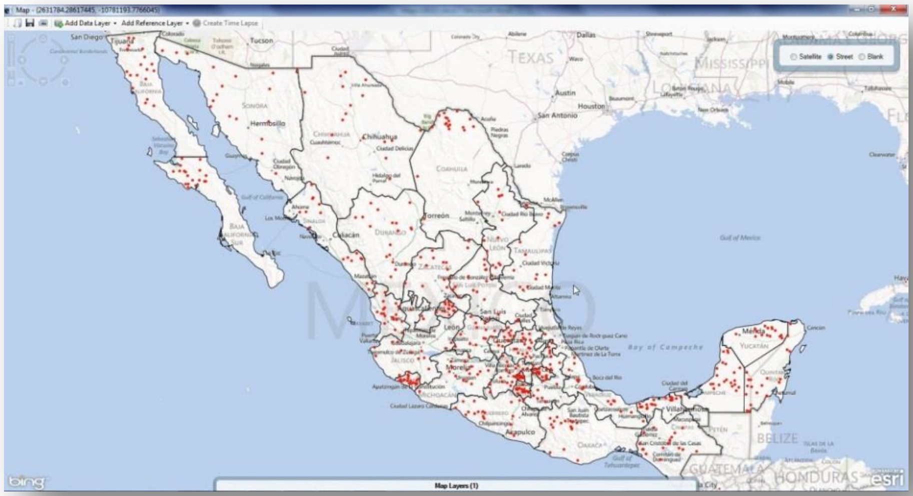


Figure 10.62: Dot Density Map in Street View

12. Select **Blank** view to display the dot density map with a white background.

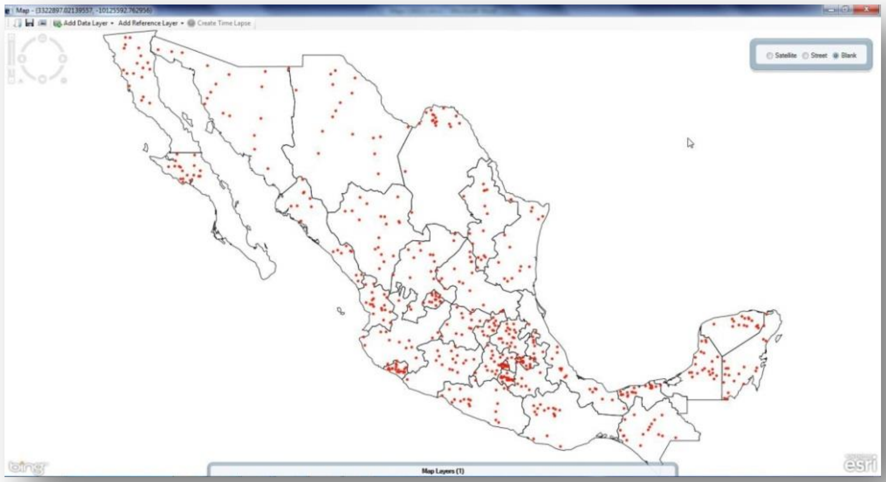


Figure 10.63: Dot Density in Blank View

13. To edit the color scheme or add a filter, place the mouse over the Map Layers gadget located at the bottom of the screen.

The three examples shown in the Adding a Data Layer section do not represent every possible combination of map type, database type and boundary file. The Epi Info™ Maps tool is capable of mapping all combinations using similar processes.

Reference Layer

Reference layers are available to enhance the map with additional information outside of the project data. In the previous section, you combined information from shape files, KML files, and the map server with your dataset to create choropleth and dot density maps. However, additional layers may be necessary to highlight points of interest or boundaries that are not evident in the data layers.

Map Server

1. From the toolbar at the top of the Map window, select **Add Reference Layer > Map Server**. The **Map Server** dialog box appears.

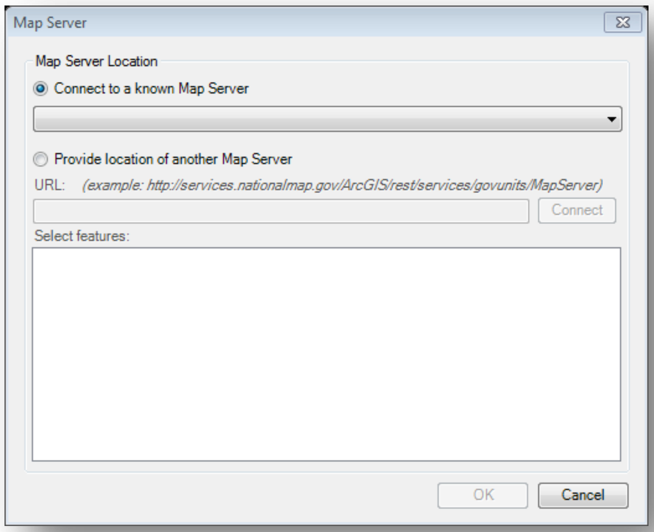

Map Server
Map Server Location
Connect to a known Map Server
Provide location of another Map Server
URL: (example: http://services.nationalmap.gov/ArcGIS/rest/services/govunits/MapServer)
Connect
Select features:
OK
Cancel

Figure 10.64: Map Server Location dialog box

2. Select the desired reference layer from the **Connect to a known Map Server** drop-down list or **Provide the location of another Map Server**. For this example, select **USGS.gov (National Hazards Support System) – Current Earthquakes** from the drop-down list. Click **OK**.

3. The map displays the location of all current earthquakes tracked in USGS' map server.

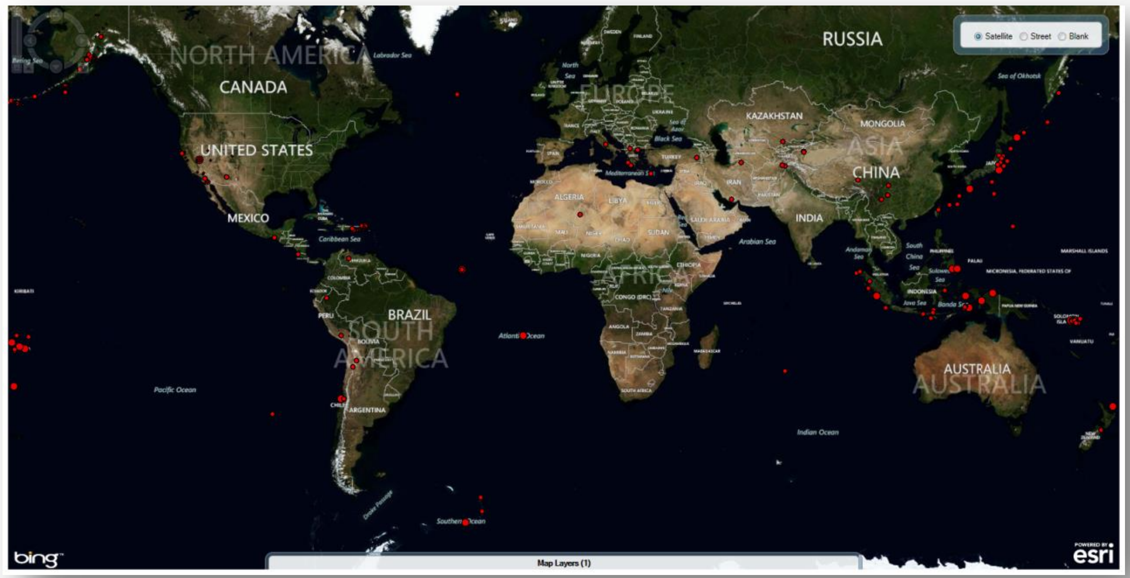


Figure 10.65: Map Server Reference Layer

Shape Files

The example below displays the Mexican state boundaries from the Sample project folder.

1. From the toolbar at the top of the map window, select **Add Reference Layer > Shape Files**.
2. Browse and select the desired shape file document, **Sample>MXState.shp**. The map appears displaying the shape file boundaries.

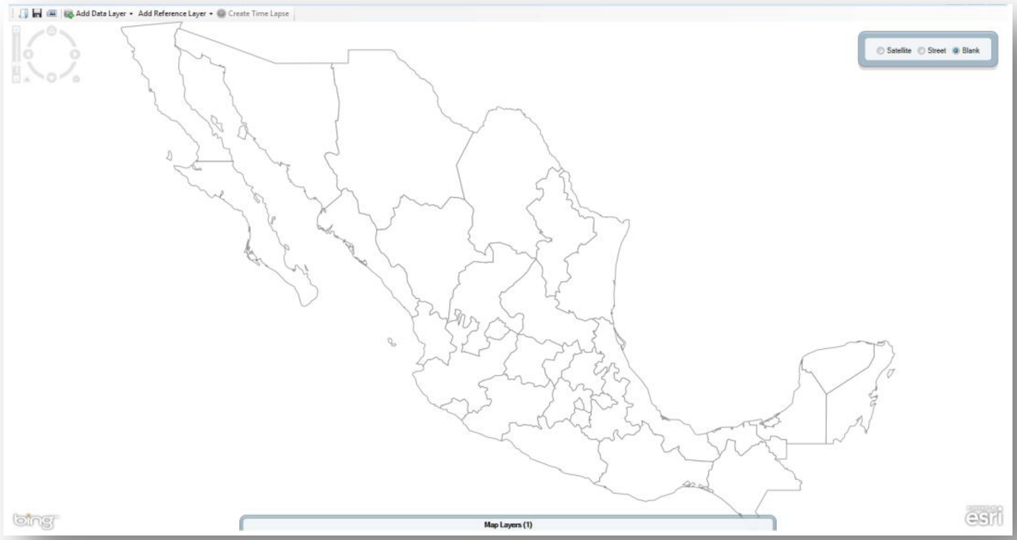


Figure 10.66: Shape file Reference Layer

KML

The example below displays the storm track taken by Hurricane Katrina.

1. From the toolbar at the top of the Map window, select **Add Reference Layer > KML**.
2. Type the following URL for NOAA's Katrina tracker: http://ngs.woc.noaa.gov/storms/katrina/kml/katrina_track.kml. The map appears displaying Hurricane Katrina's track.

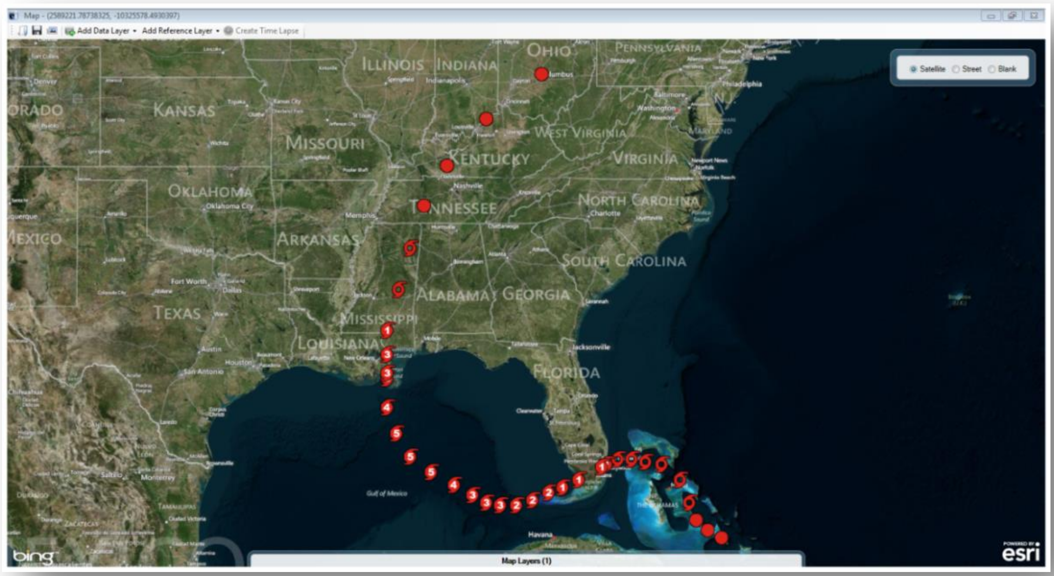


Figure 10.67: KML Reference Layer

THIS PAGE IS

INTENTIONALLY BLANK.

11. Nutritional Anthropometry: Collecting and Analyzing Nutritional Data

Introduction

Epi Info™ 7 contains several nutritional anthropometry tools used to collect, analyze, and graph child growth data. Formerly referred to as the NutStat module, the Nutritional Anthropometry tools now include a data entry form that calculates z-scores and percentiles as data are entered, creates growth charts in Visual Dashboard, and has the ability to add z-scores and percentiles to existing datasets.

Four growth references are available for use: CDC 2000 Growth Reference, WHO Child Growth Standards, WHO Reference 2007, and CDC/WHO 1978. Except for the Nutrition project forms, all nutritional anthropometry tools in Epi Info™ 7 can easily select the required growth reference.

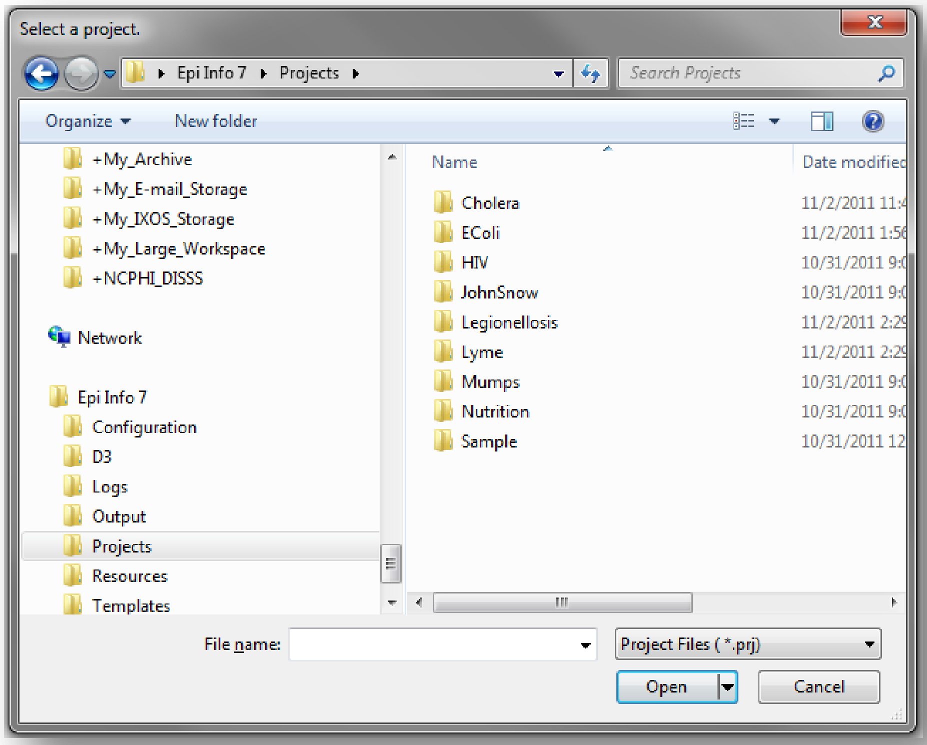
The Nutrition Project

The Nutrition Project is a specialized project that can be opened in the Enter tool or modified in Form Designer. The Nutrition form contains fields that collect a child's age, height, weight, and other measurements for each clinic or doctor's office visit. Based on these measurements, the appropriate z-scores and percentiles are automatically calculated and added to the dataset. The WHO Child Growth Standards are used to calculate z-scores for children 0 to 2 years of age and the CDC 2000 reference is used to calculate z-scores for children 2 years of age and older. For additional information, see the CDC Growth Charts Web page. (www.cdc.gov/growthcharts).

Opening the Nutrition Form

To open the Nutrition project:

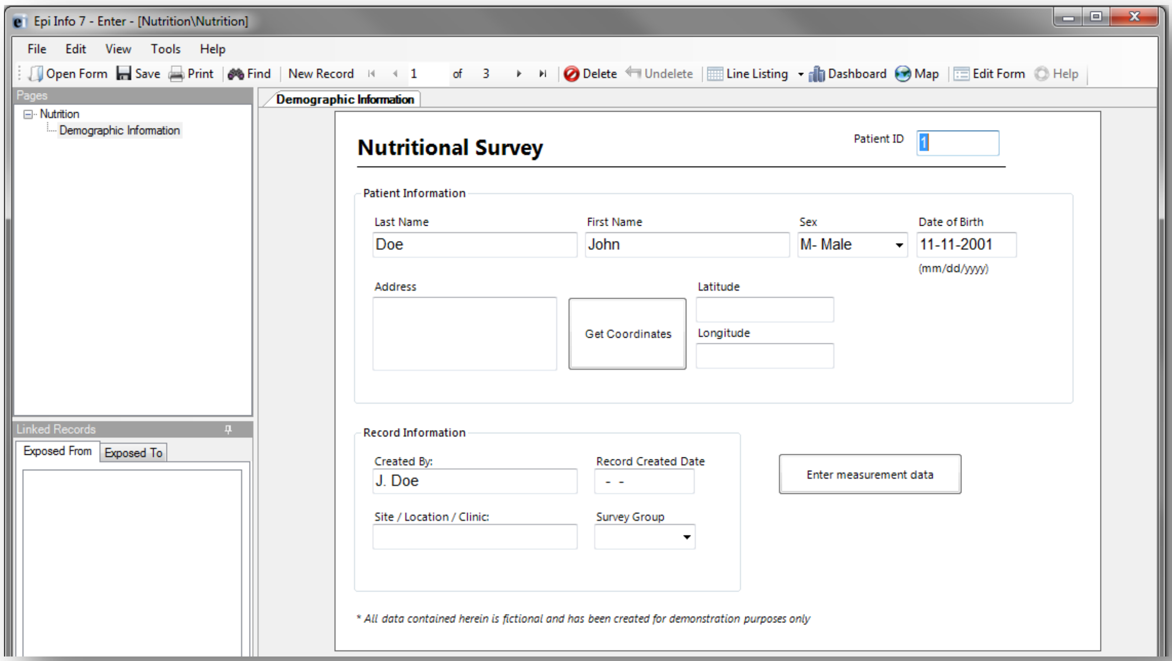
1. From the Epi Info™ 7 menu, click the **Enter Data** button or **Tools > Enter Data**. The Enter Data tool opens.
2. From the toolbar, click the **Open Form** button. The **Open Form** dialog box appears.
3. Click the **ellipsis** next to the **Current Project** text box. The **Select a Project** dialog box appears.
4. Navigate to the Projects subfolder. A list of projects appears.



Name	Date modified
Cholera	11/2/2011 11:4
EColi	11/2/2011 1:56
HIV	10/31/2011 9:0
JohnSnow	10/31/2011 9:0
Legionellosis	11/2/2011 2:29
Lyme	11/2/2011 2:29
Mumps	10/31/2011 9:0
Nutrition	10/31/2011 9:0
Sample	10/31/2011 12

Figure 10.68: Project's subfolder

5. Select the **Nutrition** folder. Click **Open**.
6. Select the **Nutrition** project file. Click **Open**. The **Open Form** dialog reappears with a list of available forms to select.
7. From the list of forms, select **Nutrition**.
8. Click **OK.** The Nutrition form appears in the Enter Data tool.



Nutritional Survey

Patient ID: 1

Patient Information

Last Name	First Name	Sex	Date of Birth
Doe	John	M- Male	11-11-2001 (mm/dd/yyyy)

Address:

Latitude:

Longitude:

Record Information

Created By:	Record Created Date
J. Doe	- -

Site / Location / Clinic:

Survey Group:

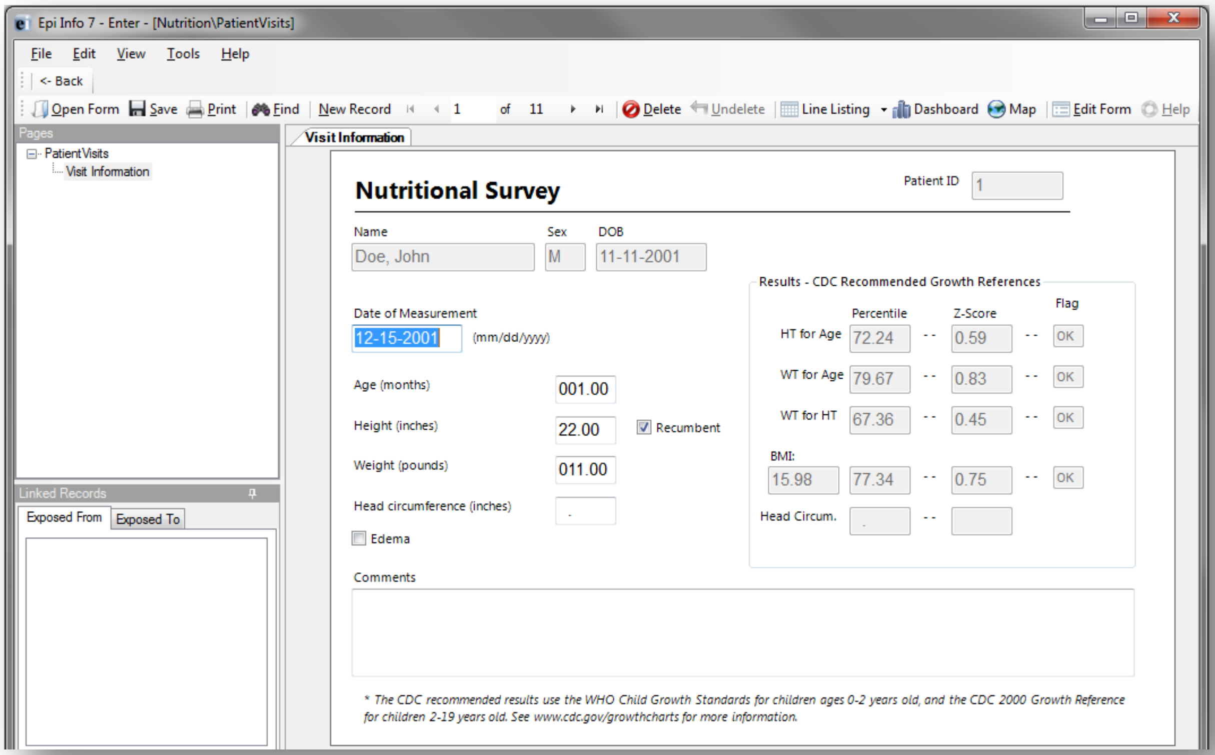
* *All data contained herein is fictional and has been created for demonstration purposes only*

Figure 10.69: The Nutrition Form

Using the Nutrition Form

After the Nutrition Form is opened, data can be entered, modified, and deleted in the same manner as other Epi Info™ 7 forms. Each record in the Nutrition form represents a single child. The form contains the optional capability to geocode the child's home address into latitude and longitude coordinates with the **Get Coordinates** button. Reference the Enter Data section of this user guide for additional information regarding the data entry process.

Although the data entry form contains only one record per child, each child may have been measured at multiple points in time. The Nutrition project has the ability to capture multiple measurements for each child, which is essential for calculating statistics and generating graphs. The Nutrition form has a related form called **PatientVisits** to store measurement information over time. The PatientVisits form may have one or more associated records. Clicking the **Enter Measurement Data** button will access the PatientVisits form.


Epi Info 7 - Enter - [Nutrition\PatientVisits]
File Edit View Tools Help
<- Back
Open Form Save Print Find New Record 1 of 11 Delete Undelete Line Listing Dashboard Map Edit Form Help
Pages
PatientVisits
Visit Information
Linked Records
Exposed From Exposed To
Visit Information

Nutritional Survey

Patient ID: 1

Name: Doe, John
Sex: M
DOB: 11-11-2001

Date of Measurement: 12-15-2001 (mm/dd/yyyy)

Age (months): 001.00

Height (inches): 22.00 [✓ Recumbent]

Weight (pounds): 011.00

Head circumference (inches): .

Edema

Comments

Results - CDC Recommended Growth References

		Percentile		Z-Score		Flag
HT for Age		72.24	--	0.59	--	OK
WT for Age		79.67	--	0.83	--	OK
WT for HT		67.36	--	0.45	--	OK
BMI:	15.98	77.34	--	0.75	--	OK
Head Circum.		.	--			

* The CDC recommended results use the WHO Child Growth Standards for children ages 0-2 years old, and the CDC 2000 Growth Reference for children 2-19 years old. See www.cdc.gov/growthcharts for more information.

Figure 10.70: PatientVisits Form

The patient's name, sex, date of birth, and ID number automatically appear in the form. These values link the records in this form to the child's nutritional record in the Nutrition form. As you type values into the PatientVisits form (e.g., date of measurement, age in months, height, weight, etc.) the z-scores and percentiles are automatically calculated and appear in the form.

Add measurement records by clicking the **New Record** button on the toolbar. Clicking the **<- Back** button closes the PatientsVisits form and returns you to the Nutrition form.

The relationship between these two forms defines how information is stored and calculated. For each record in the Nutrition form, there may be one or more linked record(s) in the PatientVisits form. If the user is viewing record #1 in the Nutrition form, clicking the **Enter Measurement Data** button will take the user to all of the measurements associated with that particular child (and no others). See the following figure for a visual explanation of the relationship. Note that the blue squares represent records in the Nutrition form, and the red squares represent records in the PatientVisits form.

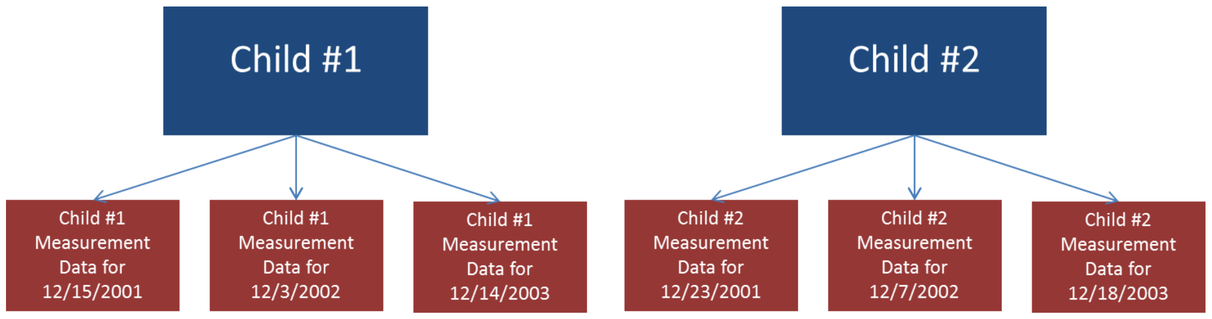

Child #1
Child #2
Child #1 Measurement Data for 12/15/2001
Child #1 Measurement Data for 12/3/2002
Child #1 Measurement Data for 12/14/2003
Child #2 Measurement Data for 12/23/2001
Child #2 Measurement Data for 12/7/2002
Child #2 Measurement Data for 12/18/2003

Figure 10.71: Nutrition (blue, top) and PatientVisits form (red, bottom) Relationship

To view Child #1's measurement information on 12/15/2001, navigate to Child #1's record in the Nutrition form. Click **Enter Measurement Data** and navigate to the record that has 12/15//2001 as the date of measurement.

Note: The data included in the Nutrition project may differ from that displayed in the figure above.

1.1.1 Customizing the Nutrition Forms

The Nutrition forms can be customized to include or exclude any field or check code in the form. For example, if geocoding the child's address into latitude and longitude is not necessary, those fields can be deleted. The form's title, Nutritional Survey, can also be changed.

Note: Removing fields or changing field names may disable the underlying z-score and percentile calculations.

For additional information regarding form editing, reference the Form Designer section of the user guide.

NutStat Growth Charts in Visual Dashboard

Based on a child's measurements, a growth chart can be created in Visual Dashboard to track the progress of a particular child (Displaying multiple children on a single chart is not an available option). The NutStat growth charts can be accessed in Visual Dashboard by right clicking on the canvas and then selecting **Add Nutstat growth chart**.

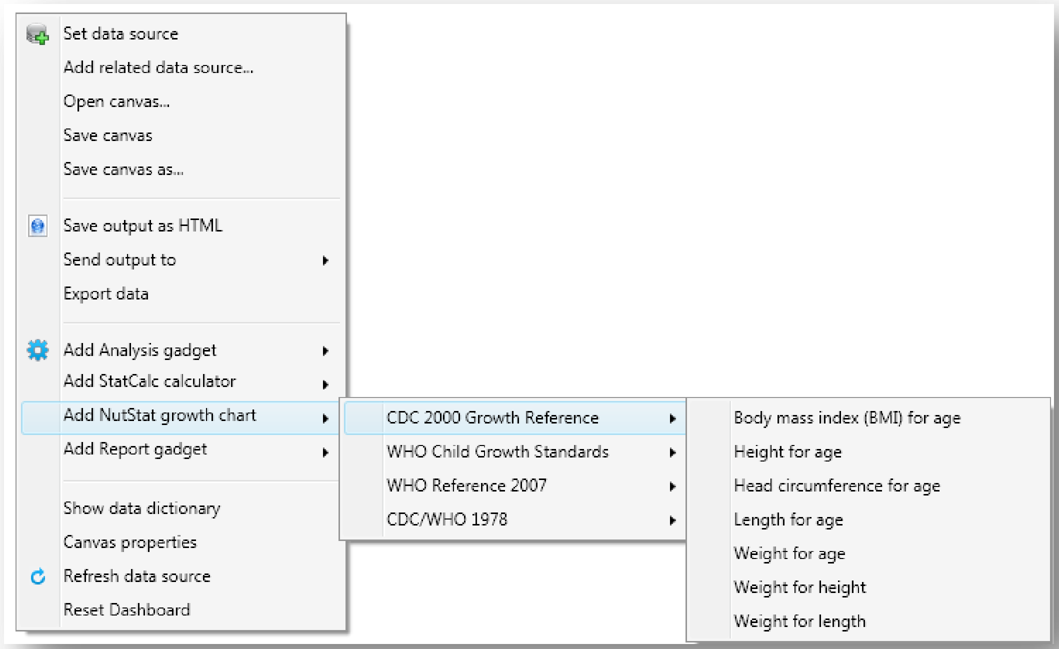


Figure 10.72: Growth Chart Types

There are four growth references available for use: CDC 2000 Growth Reference, WHO Child Growth Standards, WHO Reference 2007, and CDC/WHO 1978. Each growth reference contains different options for creating a growth chart and each chart requires the associated data.

Note: The Nutrition project does not contain example data to create all chart types.

Loading Nutritional Data

Prior to creating a growth chart, nutritional data must be loaded into the dashboard. Follow the steps below to load the data associated with the Nutrition project into the dashboard.

1. From the Epi Info™ main menu, click the **Visual Dashboard** button or select T**ools > Analyze Data > Visual Dashboard**. The Dashboard tool appears.
2. Right-click on the **dashboard canvas**. A context menu appears.
3. Select **Set Data Source** from the context menu. The **Select Data Source** dialog box appears.
4. Click the **ellipses** next to the **Data Source** text box. A file open dialog box appears.
5. Navigate to the Epi Info™ projects folder.
6. Open the **Nutrition** folder within the projects folder.
7. Select the **Nutrition** project. Click **Open**.
8. In the **Select Data Source** dialog box, select the **PatientVisits** form. Click **OK**. The data is loaded on the dashboard.

After the PatientVisits form is loaded, the dashboard will have cached the data in the computer's memory. Based on the data, various types of statistical tools or gadgets can be added to the canvas to display useful information to the user. For information regarding the statistical tools and gadgets available, reference the Visual Dashboard section of this user guide.

Creating a Growth Chart

There are many growth charting options available under **Add NutStat growth chart** in Visual Dashboard. The available growth chart types are dependent on each growth reference and require the necessary project data.

Note: The Nutrition project does not contain example data for every growth chart type.

The following examples demonstrate how to create a growth chart with data contained in the Nutrition project. All growth charts created in Visual Dashboard use similar processes outlined in the examples below.

BMI for Age

The following example demonstrates how to create a BMI for Age growth chart using the CDC 2000 Growth Reference. The growth chart will demonstrate how the patient's BMI compares to the CDC 2000 Growth Reference by age.

1. Right-click on the **dashboard canvas**. A context menu appears.
2. Select **Add NutStat Growth Chart** > **CDC 2000 Growth Reference** > **Body Mass Index (BMI) for Age** from the list of options in the context menu. A growth chart gadget appears on the canvas.

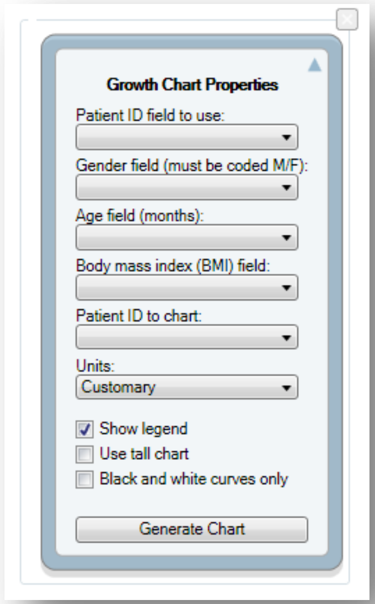
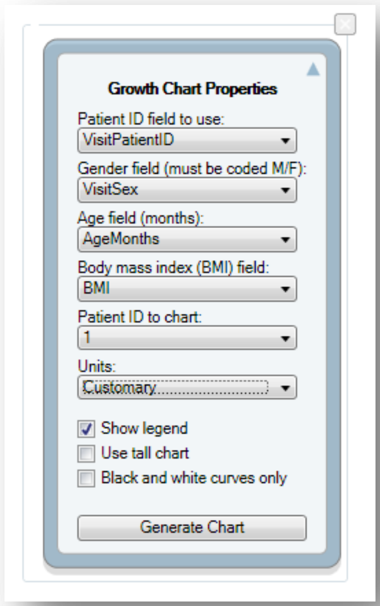


Figure 10.73: Growth Chart Gadget

The gadget must specify what fields to use for the **Patient ID**, **Gender, Age, Body mass index (BMI)**, and the **Patient ID** to chart.

3. Select **VisitPatientID** for **Patient ID field to use**.
4. Select **VisitSex** for **Gender field (must be coded M/F)**.
5. Select **AgeMonths** for **Age field (months).**
6. Select **BMI** for **Body mass index (BMI) field**.
7. Select **1** for **Patient ID to chart**.


Growth Chart Properties

Patient ID field to use: VisitPatientID

Gender field (must be coded M/F): VisitSex

Age field (months): AgeMonths

Body mass index (BMI) field: BMI

Patient ID to chart: 1

Units: Customary

- [✓] Show legend
- [ ] Use tall chart
- [ ] Black and white curves only

Figure 10.74: Completed Growth Chart Gadget

The checkboxes below the drop-down lists allow for customizing the appearance of the graph.

- **Show legend** – adds a legend to the graph (checked by default)
- **Use tall chart** – extends the vertical height of the growth chart
- **Black and white curves only** – all curves appear in black and white (color is removed)

8. Click the **Generate Chart** button. The growth chart appears on the canvas.

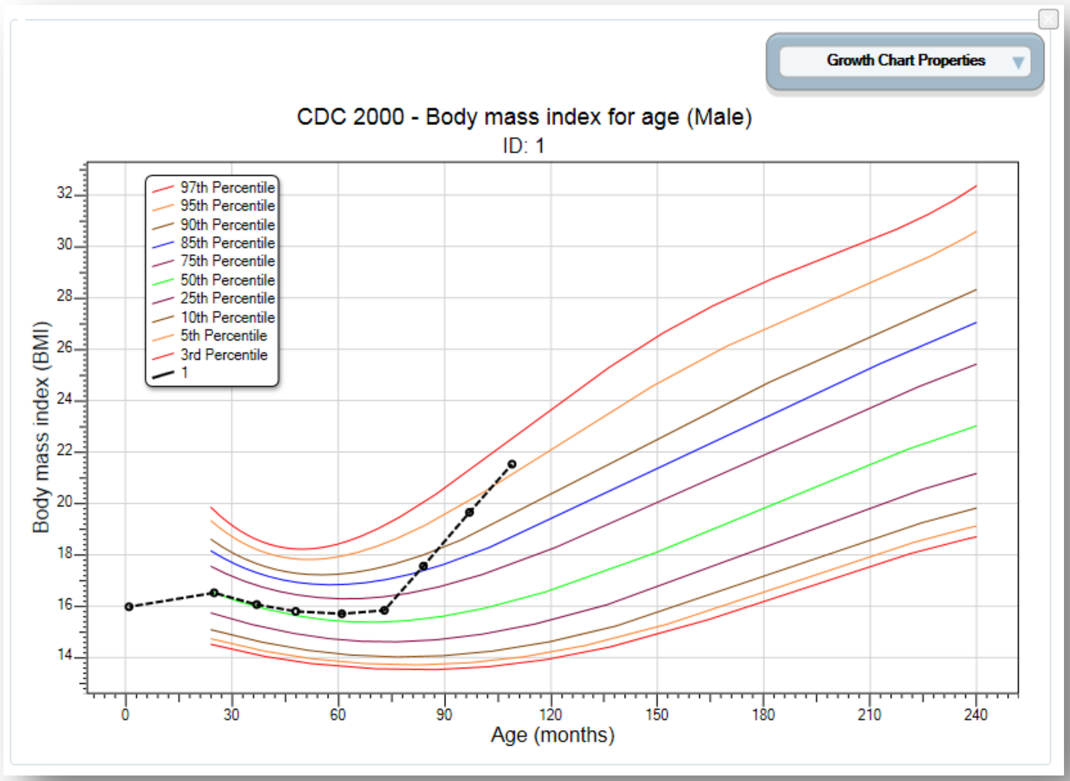


Figure 10.75: BMI for Age Growth Chart

9. To collapse the panel and show only the chart, click the **up arrow** at the top right corner of the properties panel. Click the **down arrow** to expand the properties panel.

Height for Age

The following example demonstrates how to create a Height for Age growth chart using the CDC/WHO 1978 growth reference. The growth chart will demonstrate how the patient's height compares to CDC/WHO 1978 standards by age.

1. Right-click on the **dashboard canvas**. A context menu appears.
2. Select **Add NutStat Growth Chart** > **CDC/WHO 1978**> **Height for Age** from the list of options in the context menu. A growth charting gadget appears on the canvas.
3. Select **VisitPatientID** for **Patient ID field to use**.
4. Select **VisitSex** for **Gender field (must be coded M/F).**
5. Select **AgeMonths** for **Age field (months).**
6. Select **HeightInches** for **Height field**.
7. Select **1** for **Patient ID to chart**.
8. Click the **Generate Chart** button. The chart appears as shown below.

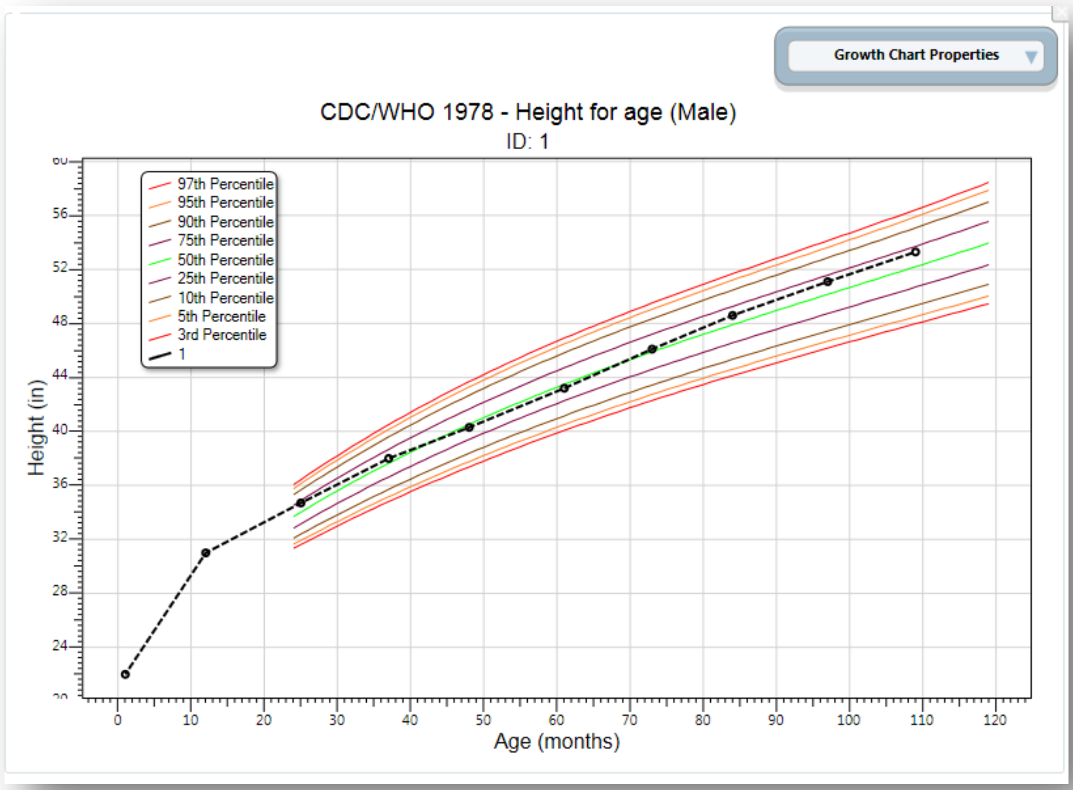


Figure 10.76: Height for Age Growth Chart

9. To collapse the panel and show only the chart, click the **up arrow** at the top right corner of the properties panel. Click the **down arrow** to expand the properties panel.

Length for Age

The following example demonstrates how to create a Length for Age growth chart using the WHO Child Growth Standards. The growth chart will demonstrate how the patient's length compares to WHO Child Growth Standards by age.

Note: The Length for Age chart requires similar inputs as the Height for Age chart but additionally requires the user to check the Recumbent check box in the PatientVisits form during data entry.

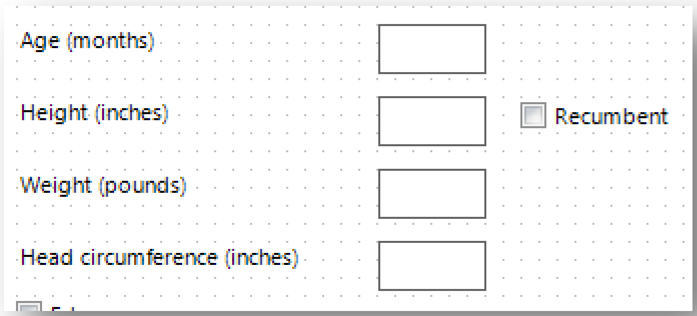


Figure 10.77: Recumbent checkbox in PatientsVisit form during data entry

1. Right-click on the **dashboard canvas**. A context menu appears.
2. Select **Add NutStat Growth Chart** > **WHO Child Growth Standards** > **Length for Age** from the list of options in the context menu. A growth charting gadget appears on the canvas.
3. Select **VisitPatientID** for **Patient ID field to use**.
4. Select **VisitSex** for **Gender field (must be coded M/F)**.
5. Select **AgeMonths** for **Age field (months)**.
6. Select **HeightInches** for **Length field**. (If recumbent was checked during data entry, the height field converts to length)
7. Select **1** for **Patient ID to chart**.
8. Click the **Generate Chart** button. The chart will appear as shown below.

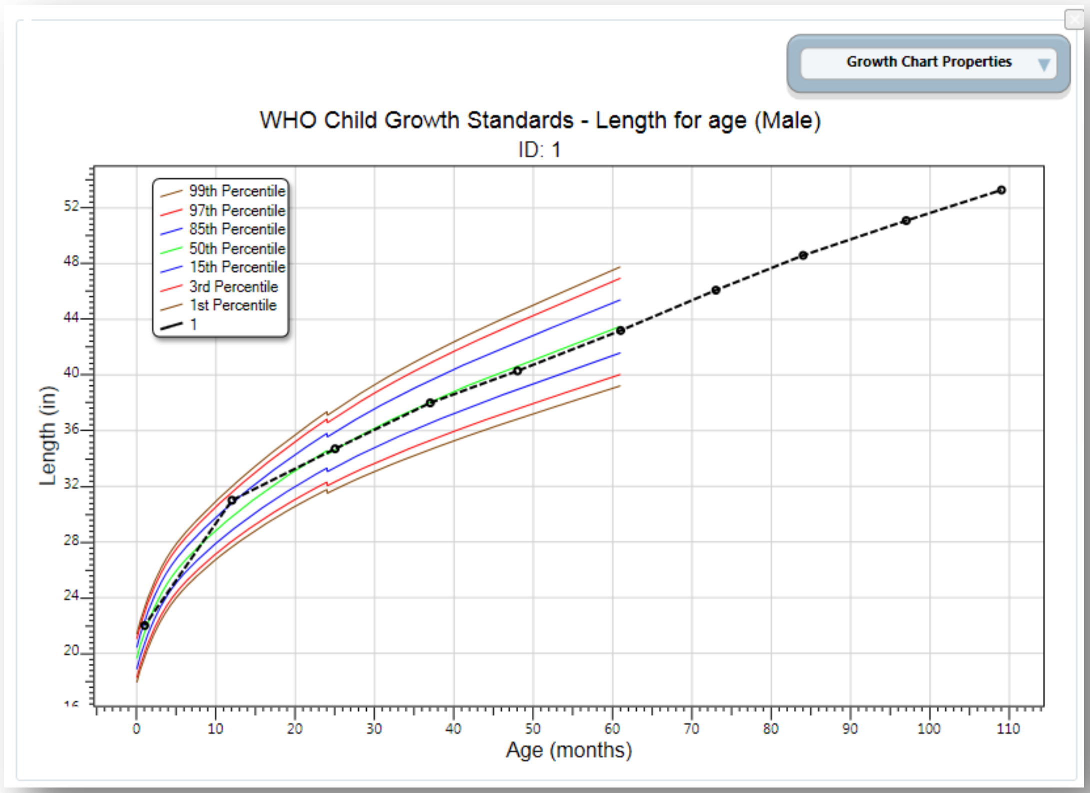


Figure 10.78: Length for Age Growth Chart

9. To collapse the panel and show only the chart, click the **up arrow** at the top right corner of the properties panel. Click the **down arrow** to expand the properties panel.

It is important to note some aspects of the growth charting capabilities. First, there must be a patient ID field to identify a single person within the database. In the examples above, all of the records from the patient with an ID value of 1 were charted. Second, the field storing gender information must code that information as M and F. The dashboard has data recoding capabilities that can be used to change gender information to other formats. No matter what format represents male and female, it can be converted. In the examples above, no conversion was necessary. Third, the field storing the age values must be in months. Similar to gender, the dashboard can convert ages stored in days and years into months. Fourth, all properties can be changed in order to re-generate the chart. However, changing the chart inputs and growth references is subject to the data available and the available chart types. Finally, the growth charts can be used with non-Epi Info™ 7 projects, including projects that do not have the z-scores and percentiles already calculated.

Using the Nutrition Functions

There may be cases where nutritional data have already been collected in another program (e.g., Microsoft Excel) but the z-scores and percentiles are missing. Re-entering all of this data into the Epi Info™ 7 Nutrition project would be inefficient and time-consuming.

The Epi Info™ 7 Classic Analysis tool contains two specialized functions, ZSCORE and PFROMZ, that can add z-scores and percentiles to an existing set of data. For additional information on the Classic Analysis tool, reference the Classic Analysis section of this user guide.

The ZSCORE Function

The ZSCORE function takes a series of parameters and returns a z-score (z statistic) based on those parameters. The nutritional parameters include:

1. Growth reference set (i.e., the CDC 2000 Growth Reference).
2. Type of z-score (i.e., body mass index for age).
3. Name of the column that stores the primary measurement (i.e., BMI).
4. Name of the column that stores the secondary measurement (i.e., AgeMonths).
5. Name of the column that stores the child's gender, stored as 1's and 2's.

Note: The data available in the Nutrition project stores gender information as M and F and must be converted to 1's and 2's.

Each parameter is separated by a comma, and the entire series of parameters are enclosed in parenthesis. If you wish to calculate body mass index for age using the CDC 2000 Growth Reference, the function must reference the child's raw BMI (column **BMI**), the child's age in months (column **AgeMonths**), and the child's gender (column **Gender**). The function will be similar to the following:

ZSCORE("CDC 2000", "BMI", BMI, AgeMonths, Gender)

The first parameter is "CDC 2000". The quotes are required as the first two parameters are literal values (i.e. fixed values, the calculations will always refer to the exact CDC 2000 value). The full set of valid references that can be used for this parameter are listed below.

Parameter value	Reference
CDC 2000	CDC 2000 Growth References
WHO CGS	WHO Child Growth Standards (0-5 years)
WHO 2007	WHO Reference 2007 (5-19 years)
NCHS 1977	CDC/WHO 1977 Growth Reference

Figure 10.79: Valid Values for First Parameter

The second parameter, "BMI", tells the program to calculate a z-score for body mass index for age. The quotes for this second parameter are also required. The full set of valid z-score types that can be used for the second parameter are listed below. Not all growth references support all types of z-score calculations (i.e., "BMI" may be valid for CDC 2000, but it is not supported when using the NCHS 1977 growth reference).

Parameter value	Type of z-score
BMI	Body mass index-for-age
HtAge	Height-for-age
WtAge	Weight-for-age
WtHt	Weight-for-height
WtLgth	Weight-for-length
HeadCircum	Head circumference-for-age
LgthAge	Length-for-age
Ssf	Subscapular skinfold-for-age
Tsf	Triceps skinfold-for-age

Figure 10.80: Valid Values for Second Parameter

The third parameter is also BMI, but does not contain any quotes because this parameter is the name of the column (or field) in the current dataset that contains the child's raw BMI calculation. For this example, it is assumed that the raw BMI score is stored in a column called BMI.

The fourth parameter is **AgeMonths**. It is critical to note that this parameter assumes the specified column is numeric and the numbers represent the child's age in months.

The fifth parameter is **Gender**. It assumes the genders are stored in the database as 1's (male), and 2's (female).

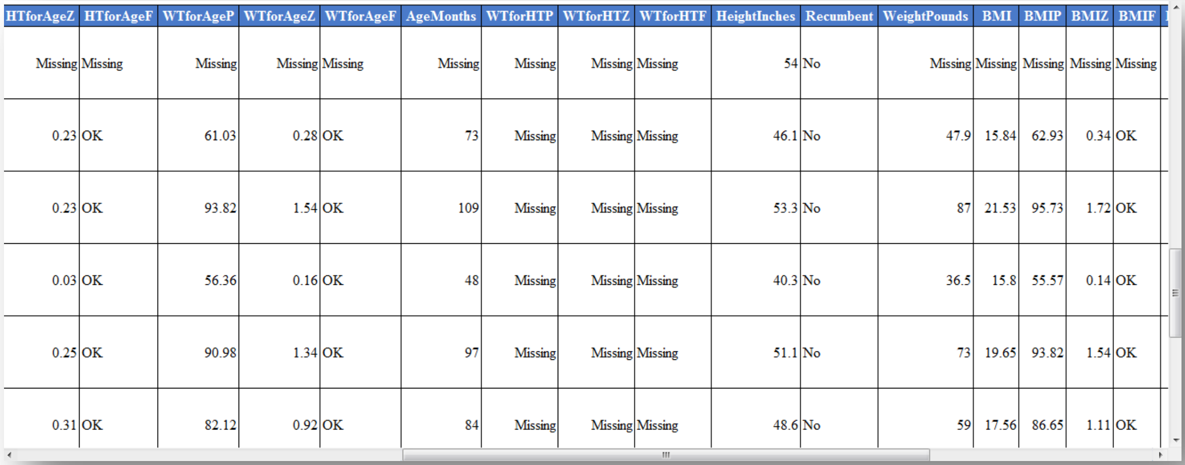
Note: The data available in the Nutrition project stores gender information as M and F under the field VisitSex. To create the necessary Gender data, enter the following code into Classic Analysis.

```
DEFINE Gender

IF VisitSex = "M" THEN
       ASSIGN Gender = "1"
END

IF VisitSex = "F" THEN
       ASSIGN Gender = "2"
END
```

Database names and column names may differ from the example above. The final three parameters may vary considerably depending on the database. However, the first two parameters will only accept the limited set of inputs stated in the tables above.

To run this calculation, the ZSCORE function must be paired with an ASSIGN command. For example:

```
ASSIGN BMIZ = ZSCORE("CDC 2000", "BMI", BMI, AgeMonths, Gender)
```



HTforAgeZ	HTforAgeF	WTforAgeP	WTforAgeZ	WTforAgeF	AgeMonths	WTforHTP	WTforHTZ	WTforHTF	HeightInches	Recumbent	WeightPounds	BMI	BMIP	BMIZ	BMIF
Missing	Missing	Missing	Missing	Missing	Missing	Missing	Missing	Missing	54	No	Missing	Missing	Missing	Missing	Missing
0.23	OK	61.03	0.28	OK	73	Missing	Missing	Missing	46.1	No	47.9	15.84	62.93	0.34	OK
0.23	OK	93.82	1.54	OK	109	Missing	Missing	Missing	53.3	No	87	21.53	95.73	1.72	OK
0.03	OK	56.36	0.16	OK	48	Missing	Missing	Missing	40.3	No	36.5	15.8	55.57	0.14	OK
0.25	OK	90.98	1.34	OK	97	Missing	Missing	Missing	51.1	No	73	19.65	93.82	1.54	OK
0.31	OK	82.12	0.92	OK	84	Missing	Missing	Missing	48.6	No	59	17.56	86.65	1.11	OK

Figure 10.81: BMIZ Calculation

The code above instructs the program to assign the z-score for body mass index for age using the CDC 2000 Growth Reference to the BMIZ column. This will be done for every row in the current dataset.

The PFROMZ Function

The second Nutritional Anthropometry function included in Epi Info™ 7 is called PFROMZ, which converts a z-score into a percentile. To perform the percentile calculation, enter the column name that contains a z-score as a parameter, and the corresponding percentile is returned. For example:

```
PFROMZ(BMIZ)
```

The above code assumes the BMIZ column contains the z-scores for body mass index for age. Note that the PFROMZ function does not need to know the reference, z-score type, gender, etc., of the data. For example:

```
ASSIGN BMIP = PFROMZ(BMIZ)
```

The code above instructs the program to take the percentile associated with the z-score values stored in the BMIZ column and to assign that value to the BMIP column. This will be done for every row in the current dataset.

12. StatCalc

Introduction

StatCalc is an epidemiologic calculator that produces statistics from summary data. Three types of calculations are offered:

- Statistics from 2-by-2 to 2-by-9 tables similar to those produced in Analysis. Both single and stratified 2-by-2 tables can be analyzed to produce odds ratios and risk ratios (relative risks) with confidence limits, several types of chi square tests, Fisher exact tests, Mantel-Haenszel summary odds ratios and chi squares, and associated p-values.
- Sample size and power calculations include Population Survey, Cohort or Cross-Sectional, and Unmatched Case-Control Study.
- Chi-square for trend by the Mantel extension of the Mantel-Haenszel summary odds ratio and chi square. This tests for the presence of a trend in dose response or other case control studies where a series of increasing or decreasing exposures is being studied.

How to Use StatCalc

Opening StatCalc

1. From the Epi Info 7 main menu, select **StatCalc**.
2. Select one of the following calculations using the up and down arrow keys or click: **Sample Size & Power, Chi Square for Trend, Tables (2X2, 2Xn), Poisson (rare event vs. std.) or Binomial (proportion vs. std.).**
    - You can also select calculations by pressing the corresponding key on the keyboard that matches the highlighted letter on the menu. If you select a calculation using the keyboard, the calculation appears immediately.
    - If you select the Sample size and power calculation, the following calculation options appear: Population Survey, Cohort or Cross-Sectional, Unmatched Case-Control.
    - Use the tab key or return key to move around the different cells available for data entry

4. Enter data for each calculation type. Calculations are performed when you enter data...
5. To modify values already entered, use the **Tab key** or click on the **cell** and enter the new information.
    - Right click and select the **Print** option in order to print the results.
    - Right click and select the **Save as Image** option to save a screen shot of your results as an image file.

Analysis of Single and Stratified Tables

These tables are similar to those produced in Classic Analysis. Both single and stratified 2-by-2 tables can be analyzed to produce odds ratios and risk ratios (relative risks) with confidence limits, several types of chi square tests, Fisher exact tests, Mantel-Haenszel summary odds ratios and chi squares, and associated p values

Assumptions

- The values in the cells must be counts representing the number of records meeting the specifications in the marginal and stratum labels.
- A case-control study is one in which the ill and well individuals are selected and the number of exposed and unexposed is subsequently ascertained. In a cohort study, the exposed and unexposed are selected and the number of ill in each group is subsequently ascertained. A cross-sectional study starts with neither illness nor exposure determined, and ascertains both during the study.
- In cohort studies, the relative risk may be calculated from the results. In case-control studies, the odds ratio may be used as an approximation of the relative risk if the disease is rare in the general population from which cases and controls are selected. Fewer than one case in 20 individuals might be taken as a starting point. Thus, in a foodborne outbreak, selecting cases and controls from those who ate at a particular restaurant in a given week, the odds ratio could not be used to approximate relative risk if half of the individuals became ill. The odds ratio would, however, be an indicator of the degree of association between illness and the consumption of a particular food.
- For the results to be valid, the outcomes in each record must be independent of those in other records. The values for one individual do not predict those for another. Confounding must be removed by stratifying on confounding variables.Single 2-by-2 Tables

Two-by-two tables are frequently used in epidemiology to explore associations between exposure to risk factors and disease or other outcomes. The table in StatCalc has Exposure on the left and Disease across the top. Not all textbooks and articles use the same conventions. Carefully observe the labels and transpose data items if necessary.

Given a yes-no or other two-choice question describing disease and another describing exposure to a risk factor, StatCalc produces several kinds of statistics that test for relationships between exposure and disease. Generally, an association is suggested by an odds ratio or relative risk larger or smaller than 1.0. The further the odds ratio or relative risk is from 1.0, the stronger the apparent association. Statistical significance can be assessed by p-values for the Chi square tests that are small <.05 if often used, Fisher exact test with a small p value; or confidence limits for the odds ratio that do not include 1.0.

The expected value of a cell is the product of the marginal totals for that cell divided by the grand total for the table. If any expected value is less than five, it is recommended you use the Fisher Exact test results and the Exact confidence limits. If the numbers in the table are all large, the other tests should indicate nearly the same result.

Stratified Analysis of 2-by-2 Tables

If confounding is present, associations between disease and exposure can be missed or falsely detected. A confounding factor is one that is associated with the disease and the exposure, but may not be of interest or observed. Age is a frequent confounder, although any factor other than the main exposure being considered can be treated as a confounder.

Stratification means making a separate table of disease by exposure for each possible confounder combinations. In the simplest case, this could mean separate male and female tables if sex is the potential confounder. If Age, Sex, and City are confounders, separate tables would be made for each possible combination of age group, sex, and city.

The Mantel-Haenszel weighted odds ratio, relative risk, summary Chi square, and p-value combine results from different strata to remove confounding caused by the variables used for stratification. If tables are entered for male and female, confounding by sex is removed. The degree of confounding can be judged by comparing the crude and weighted odds ratios; if they are identical, there was no confounding by sex.

The Approximate (Cornfield and Greenland/Robins) and Exact confidence limits provide additional measures. If the weighted odds ratio or relative risk (not for case-control studies) has confidence limits that do not include 1.0, there is a statistical association with 95% confidence between the disease and the exposure without confounding by the stratifying factor.

If the odds ratio or relative risks for strata in a series of stratified tables for the tests are not similar, then interaction between the stratifying factor and the risk factor are present. Logistic regression or other multivariate methods may be indicated (or the number of subjects is too small to draw definite conclusions). Alternatively, it may be advisable to present the stratum-specific estimates separately.

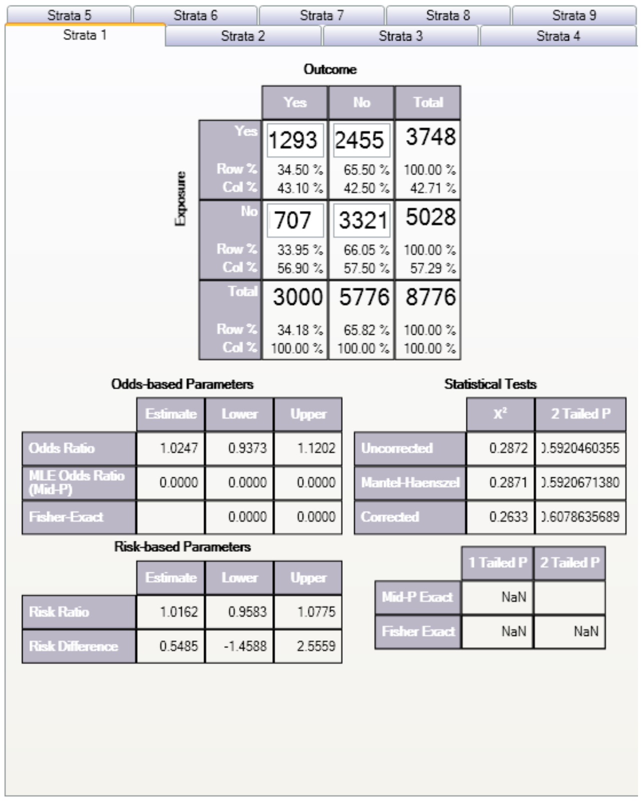
Experts in logistic regression recommend you thoroughly explore the data using stratified analysis before undertaking the regression analysis. If the number of confounding factors is fairly small and the odds ratios are homogeneous from stratum to stratum, stratified analysis may be all you need.

Example

A study of bladder cancers cases used randomly selected controls to explore the history of artificial sweetener use in the two groups.

	Bladder Cancer	
Sweetener	**Yes**	**No**
Ever Used	1293	2455
Never Used	1707	3321

1. From the StatCalc application main page, select **Tables <2 x 2, 2 x n>**. A blank table opens in the StatCalc application window.
2. Enter the **data** from the above sample table.
3. Results will be generated as the different values in the cells are populated.



		Yes	No	Total
Yes		1293	2455	3748
	Row %	34.50 %	65.50 %	100.00 %
	Col %	43.10 %	42.50 %	42.71 %
No		707	3321	5028
	Row %	33.95 %	66.05 %	100.00 %
	Col %	56.90 %	57.50 %	57.29 %
Total		3000	5776	8776
	Row %	34.18 %	65.82 %	100.00 %
	Col %	100.00 %	100.00 %	100.00 %

Odds-based Parameters

	Estimate	Lower	Upper
Odds Ratio	1.0247	0.9373	1.1202
MLE Odds Ratio (Mid-P)	0.0000	0.0000	0.0000
Fisher-Exact		0.0000	0.0000

Statistical Tests

	X^2	2 Tailed P
Uncorrected	0.2872	).5920460355
Mantel-Haenszel	0.2871	).5920671380
Corrected	0.2633	).6078635689

Risk-based Parameters

	Estimate	Lower	Upper
Risk Ratio	1.0162	0.9583	1.0775
Risk Difference	0.5485	-1.4588	2.5559

	1 Tailed P	2 Tailed P
Mid-P Exact	NaN	
Fisher Exact	NaN	NaN

Figure 12.1: Results Table

The odds ratio of 1.02 and the confidence limits that include 1.0 fail to provide evidence of any association between sweetener use and bladder cancer (cited by Schlesselman, p. 38-39).

Example

Relationship Between Alcohol Consumption and Myocardial Infarction (MI): Confounding Due to Smoking Hypothetical Data from Schlesselman, p. 182

The following case-control study indicates an apparent association between alcohol consumption and MI with an odds ratio of 2.26.

	MI	
Alcohol	**Yes**	**No**
Yes	71	52
No	29	48

Smoking is known to be associated with MI and alcohol consumption. Stratifying the data by smoking status creates two tables, one for smokers, and one for nonsmokers.

Nonsmokers

	MI	
Alcohol	**Yes**	**No**
Yes	8	16
No	22	44

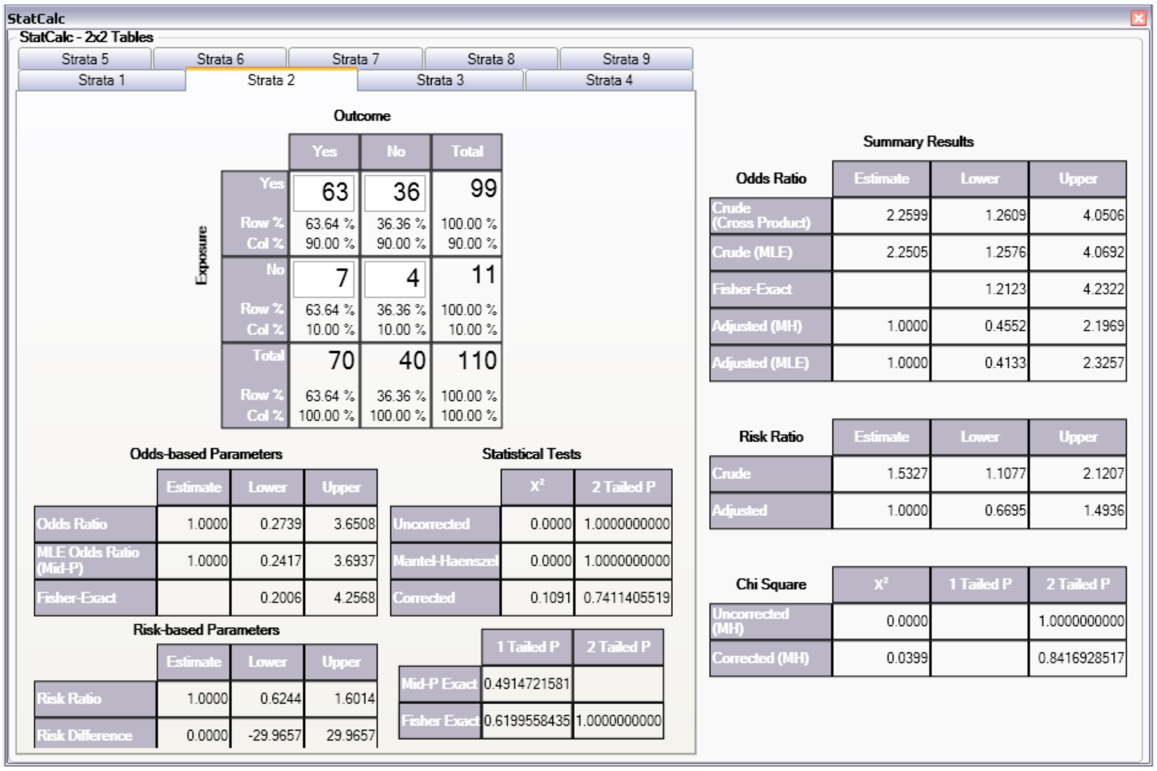
Smokers

	MI	
Alcohol	**Yes**	**No**
Yes	63	36
No	7	4

The odds ratio for each table is 1.0, and the Mantel summary odds ratio is 1.0. The crude odds ratio and the Mantel summary odds ratio are quite different (4.0 and 1.0), concluding that smoking was a confounding factor and there appears (with this over simplified analysis) to be no association (odds ratio= 1.0) between alcohol and MI. Note that the odds ratio in the two strata are the same (1.0); there is no interaction or effect modification between smoking and alcohol. In other words, the effect of alcohol on MI is the same for smokers and nonsmokers. When the effect varies in the different strata (the odds ratios are different), interaction or effect modification is present.

To view and create a Stratified Analysis Summary of 2-by-2 Tables, take the following steps:

1. From the StatCalc application main page, select **Tables <2 x 2, 2 x n>.** A blank table opens in the StatCalc application window.
2. Enter the **data** from the above Nonsmokers sample table in the first Strata tab (Strata 1).
3. Click on **Strata 2**
4. Enter the **data** from the above Smokers sample table.
5. The Stratified Analysis Summary of 2 Tables window gets populated as you enter data.



Outcome

Exposure		Yes	No	Total
Yes		63	36	99
	Row %	63.64 %	36.36 %	100.00 %
	Col %	90.00 %	90.00 %	90.00 %
No		7	4	11
	Row %	63.64 %	36.36 %	100.00 %
	Col %	10.00 %	10.00 %	10.00 %
Total		70	40	110
	Row %	63.64 %	36.36 %	100.00 %
	Col %	100.00 %	100.00 %	100.00 %

Odds-based Parameters

	Estimate	Lower	Upper
Odds Ratio	1.0000	0.2739	3.6508
MLE Odds Ratio (Mid-P)	1.0000	0.2417	3.6937
Fisher-Exact		0.2006	4.2568

Risk-based Parameters

	Estimate	Lower	Upper
Risk Ratio	1.0000	0.6244	1.6014
Risk Difference	0.0000	-29.9657	29.9657

Statistical Tests

	X²	2 Tailed P
Uncorrected	0.0000	1.0000000000
Mantel-Haenszel	0.0000	1.0000000000
Corrected	0.1091	0.7411405519

	1 Tailed P	2 Tailed P
Mid-P Exact	0.4914721581	
Fisher Exact	0.6199558435	1.0000000000

Summary Results

Odds Ratio	Estimate	Lower	Upper
Crude (Cross Product)	2.2599	1.2609	4.0506
Crude (MLE)	2.2505	1.2576	4.0692
Fisher-Exact		1.2123	4.2322
Adjusted (MH)	1.0000	0.4552	2.1969
Adjusted (MLE)	1.0000	0.4133	2.3257

Risk Ratio	Estimate	Lower	Upper
Crude	1.5327	1.1077	2.1207
Adjusted	1.0000	0.6695	1.4936

Chi Square	X²	1 Tailed P	2 Tailed P
Uncorrected (MH)	0.0000		1.0000000000
Corrected (MH)	0.0399		0.8416928517

Figure 12.2: Multi-Strata 2x2 Results Table

Population Survey or Descriptive Study

Assumptions

- The sample to be taken must be a simple random or representative. A systematic sample (e.g., every fifth person on a list) is acceptable if the sample is representative. Choosing every other person from a list of couples, however, would not give a representative sample because it might select only males or only females.
- The question being asked must have a Yes-No or other two-choice answer, leading to a proportion of the population (those answering Yes) as the final result.

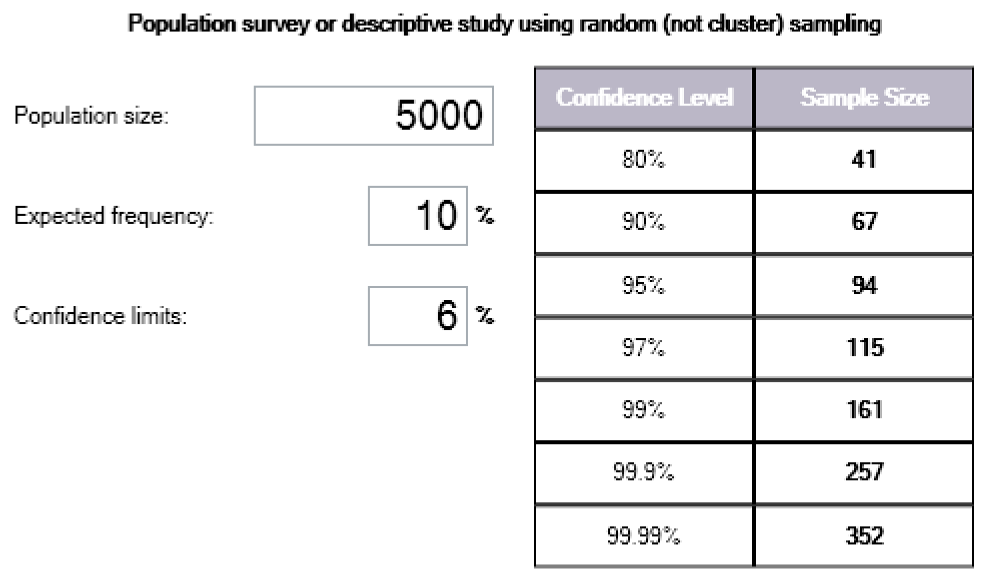
Example

Suppose you want to investigate whether the true prevalence of HIV antibody in a population is 10%. A random or systematic sample of the population is planned to estimate the prevalence. A 95% confidence that the true proportion in the entire population will fall within the confidence interval calculated from the sample is desired.

In StatCalc, enter the population size of 5,000, the estimate of the true prevalence 10%, and 10% as the Confidence Limit.Worst Acceptable value. The application will show the sample size for several different confidence levels including the desired 95%.

1. From the StatCalc application main page, select **Sample size & power.** The following calculation options appear: Population survey, Cohort or cross-sectional, Unmatched case-control.
2. Select **Population Survey**. The Population Survey or Descriptive Study Using Random (Not Cluster) Sampling window opens.
3. Enter the Population Size of **5,000**.
4. Enter the Expected Frequency of **10**%.
5. Enter the Confidence Limits as 6%.

   The results appear in the window.



Population survey or descriptive study using random (not cluster) sampling

Population size: 5000

Expected frequency: 10 %

Confidence limits: 6 %

Confidence Level	Sample Size
80%	41
90%	67
95%	94
97%	115
99%	161
99.9%	257
99.99%	352

Figure 12.3: Population Survey or Descriptive Study Results Table

Notes

Sample size determination is only a rough guide, based on assuming a specific value for the true population proportion, the variability in the sample estimate and its confidence limit. Many other factors (i.e.,cost, number of available subjects, rate of nonresponse, and the accuracy of answers and data transcription) must be considered in study design.

THIS PAGE IS

INTENTIONALLY BLANK.

13. Command Reference

Introduction

Epi Info 7 is easy to operate in interactive mode, but complex or repeated operations require saving the steps as programs. Programs (similar to "scripts" in other software) can be used to set up menus, guide and limit the data entry process, restructure data, and do analyses.

In Form Designer and Classic Analysis, programming consists of interacting with a series of dialogs that produce the actual program statements. Experienced users may want to edit the statements or type them directly in the Program Editor. For this reason, the details of command syntax are provided and include a definition of each command and its operation because a single command (e.g., EXECUTE) may be found in Form Designer, Classic Analysis, and Visual Dashboard. Commands are in module based chapters with notation of differences that may exist between program implementation. Some commands are only available in one or two programs. Check Commands are saved in Form Designer and executed in Enter. Classic Analysis commands are generated, edited, executed, and may be saved with the Program Editor from Classic Analysis.

Functions and operators appear within commands and are used for common tasks (i.e., extracting a year from a date, combining two numeric values, calculating duration between two dates, or converting numbers to text and vice versa).

Check Code Commands

ASSIGN

Description

This command assigns the result of an arithmetic or string expression to a variable.

Syntax

```
ASSIGN <variable> = <expression>

ASSIGN <variable> = <defined DLLOBJECT>!<script function>({<parameters>})
```

- The <variable> represents a variable in a database or a defined variable created in a program.
- The <expression> represents any valid arithmetic or string expression.
- The <defined DLLOBJECT> represents any variable defined as a DLL object.
- The <script function> represents the name of a class or method inside the DLL or WSC that returns the desired value to be assigned.
- The <parameters> represent one or more optional function parameters to be passed into the DLL or WSC (do not include the {} or <> symbols in the code; parenthesis are required).

Comments

This command assigns the value of an expression to a variable. The variable may be a database variable in a form or Data Table, a user-defined variable created by the DEFINE command, or a system variable.

Examples

Example 1: The patient's age is calculated using the date of birth and the date the survey was last updated. The example assumes a form exists with the following fields: DOB (Date), SurveyDate (Date), and Age (Numeric). The code below would appear in the AFTER section of the second date field to be filled in by the user.

```
IF NOT BirthDate = (.) AND NOT SurveyDate = (.) THEN
   ASSIGN Age = YEARS(BirthDate, SurveyDate)
END-IF
```

Example 2: The code below will automatically set the checkbox field 'Minor' to true when the value of the field 'Age' is below 18. The example assumes a form exists that has the following fields: Minor (Checkbox) and Age (Numeric). The code below would appear in the AFTER section of the Age field.

```
IF Age < 18 THEN
   ASSIGN Minor = (+)
END
```

Example 3: A field is assigned the value of a mathematical expression that is used to calculate body mass index. The example assumes a form exists with the following fields: BMI (Numeric), Weight (Numeric), and Height (Numeric). The code below would appear in the AFTER section of the last field to be entered. Note that Weight and Height in this formula are being measured in pounds and inches, respectively.

```
ASSIGN BMI = (Weight / (Height * Height)) * 703
```

Example 4: A patient ID field is automatically generated using the patient's last name, gender, and middle initial. The example assumes a form exists with the following fields: ID (Text), LastName (Text), Sex (Text), and MI (Text). The code below would appear in the AFTER section of the last of the above fields to be entered.

```
ASSIGN ID = LastName & " - " & Sex & " (" & MI & ")"
```

AUTOSEARCH

Description

AUTOSEARCH causes Enter to search for records with values in the specified fields that match ones in the current record. If a match is found, it can be displayed, edited, or ignored, and the current record can continue to be entered.

Syntax

```
AUTOSEARCH [<field(s)>]
```

- The <field(s)> represents one or more fields to search.

Comments

The results are displayed as a spreadsheet. If you have more records than can be viewed in a single screen, a scroll bar appears to the right of the spreadsheet. Use the mouse to see the additional matched records.

To quickly navigate to one of the matched records returned, double-click the intended **row**. Alternatively, move the cursor to the desired row and press **Enter** or click **OK**. Navigating to a matched record will discard any data entered to the new record. All fields will show data from the selected record. The record number indicator at the lower left will show the record number of the selected record. To avoid selecting any of the matched records, press **Esc** or click **Cancel** to return to the current new record. Data entry will continue for the new record.

Fields displayed from a search are determined as follows:

- If a single field is the key field, it will be displayed with as many other fields as possible.
- Multiple key fields (if any) will be displayed before any others.

Example

The AUTOSEARCH command is used to find duplicate entries during data entry. In this example, duplicates are identified by a matching first and last name as in a name-based registry system. The example assumes a form exists with the following fields: FirstName (Text) and LastName (Text). The code below would appear in the AFTER section of the second field to be filled in by the user.

```
AUTOSEARCH FirstName LastName
```

Note: When searching on multiple fields, put the AUTOSEARCH command in Check Code for a field after all the key values have been entered. In the example above, both FirstName and LastName are key fields and LastName is the last of the key fields to be entered. The AUTOSEARCH command should appear in the Check Code for LastName.

BEEP

Description

This command causes the computer to generate a beep sound. It is often used to emphasize a customized message or warning dialog during data entry.

Syntax

```
BEEP
```

Comments

Command must be typed into the Program Editor since it is not available using the Command Tree.

Example

The computer emits a beep when invalid data is detected in the Age field. This code should appear in the AFTER section of the Age field.

```
IF Age > 5 THEN
   BEEP
   DIALOG "Do not include records for children over 5."
END-IF
```

CLEAR

Description

CLEAR sets the field named to the missing value, as if it had been left blank. The command is used to clear a previous entry when an error has been detected or a change occurs. More than one field may be specified. CLEAR is frequently followed by a GOTO command, which places the cursor in position for further entry after an error.

Note: More than one field can be used with the CLEAR command.

Syntax

```
CLEAR [<field(s)>]
```

- The <field> represents the name of a field on the form. If more than one field is specified, a space will separate them.

Comments

CLEAR will delete the value only for the current record. CLEAR cannot be used in grid tables. In Classic Analysis, you must clear a variable to use the ASSIGN command to assign it a value of null (.).

Examples

Example 1: The code below prevents invalid data from being saved to the current record by erasing it as soon as it is detected. The example assumes a form exists with the following field: Age (Numeric). The code below would appear in the AFTER section of the Age field.

```
IF Age >= 18 THEN
   BEEP
   DIALOG "Do not include records for adults."
   CLEAR Age
END-IF
```

Example 2: The code below prevents an invalid date from being saved to the current record by erasing it as soon as it is detected. The example assumes a form exists with the following fields: DOB (Date) and SurveyDate (Date). The code below would appear in the AFTER section of the SurveyDate field.

```
IF (DOB > SurveyDate) OR (SurveyDate > SYSTEMDATE) THEN
   CLEAR DOB SurveyDate
   DIALOG "Invalid date detected. Please try again."
END-IF
```

Example 3: The code below prevents an invalid date from being saved to the current record by erasing it as soon as it is detected. By using a GOTO command to move the cursor back to the SurveyDate field, you are forced to keep entering data until valid data are detected. The example assumes a form exists with the following fields: DOB (Date) and SurveyDate (Date). The code below would appear in the AFTER section of the SurveyDate field.

```
IF DOB > SurveyDate THEN
   CLEAR SurveyDate
   GOTO SurveyDate
   DIALOG "Survey date invalid. Please try again."
END-IF
```

COMMENTS (*)

Description

In Check Code and Classic Analysis, the combination of backslash and asterisk in the beginning of a line of code and an asterisk and backslash at the end, as shown in some code samples, indicates a comment. Commented lines are not executed. This allows you to enter user-defined comments to identify tasks, describe variable names or code blocks for documentation, and to assist with trouble-shooting or debugging.

Syntax

```
/* <text>

*/
```

- The <text> represents any alphanumeric text as a comment or a block of code slated to be ignored.

Comments

The /* character must be placed in the first column of a line to be recognized as comment mark. For comments longer than one line, the */ characters should be added at the end of the code. Use comments to disable commands.

Examples

Example 1: Comments are used to note the date of the code's generation date, purpose, and author.

```
/* Written by Jason D. Veloper, MPH – 06/30/2010

The block of code below uses the YEARS function
*/
ASSIGN AGE = YEARS(DOB, SYSTEMDATE)
```

Example 2: Comments are used to disable certain commands from executing.

```
/* The next few lines are incomplete and are commented for later

DEFINE PatientID Numeric – Note: need to determine ID should be a

Text type variable.

LIST – ToDo: need to add the variables to list

*/
```

DEFINE

Description

This command creates a new variable. In Check Code, all user defined variables are saved in the DEFINEDVARIABLES section.

Syntax

```
DEFINE <variable> {<scope>} {<type indicator>}
```

- <variable> represents the name of the variable to be created. The name of the newly defined variable cannot be a reserved word. For a list of reserved words, see the List of Reserved Words section.
- <scope> is optional and is the level of visibility and availability of the new variable. <scope> must be one of the reserved words: STANDARD, GLOBAL, or PERMANENT. If omitted, STANDARD is assumed and a type indicator cannot be used. For information on defining a variable as a DLL OBJECT, see the DEFINE DLLOBJECT command.
- <type indicator> is required and cannot be used if <scope> is omitted. <type indicator> is the data type of the new variable and must be one of the following reserved words: NUMERIC, TEXTINPUT, YN, or DATEFORMAT. If omitted, the variable type will be inferred based on the data type of the first value assigned to the variable. However, **omitting field type is not recommended**. Once field type is defined, the variable type cannot be changed. An error will occur if you attempt to assign data of a different type to the variable.

Comments

A custom variable defined in Epi Info 7 might not have a predefined data type if the <type indicator> is omitted when the variable is defined. If the variable does not have a predefined type and has not been used, the variable will accept a value in any of the four data types (Text, Number, Date, Yes/No [Boolean]) that is assigned to the variable the first time. Thereafter, the variable takes on the data type of the value assigned and it's data type cannot be changed. However, omitting field type is not recommended. An error will occur if you attempt to assign data of a different data type. Various functions can be used to manipulate data, changing data type of values to match the data type of the variable. Some of these functions include FORMAT, TXTTONUM, TXTTODATE, and NUMTODATE.

Variable Scope

- **STANDARD** variables retain their value only within the current record and are reset when you load a new record. Standard variables are used as temporary variables behaving like other fields in the database. In Classic Analysis, Standard variables lose their values and definitions with each READ statement.

- **GLOBAL** variables retain values across related forms and when the program opens a new form, but are removed when you close the Enter program. Global variables persist while the program is executed. Global variables are also used in Classic Analysis to store values between changes of data source.
- **PERMANENT** variables are stored in the EpiInfo.Config.xml file and retain any value assigned until the value is changed by another assignment or the variable is undefined. They are shared among Epi Info programs (i.e., Enter, Classic Analysis, etc.) and persist even if the computer is shut down. Permanent variables in Classic Analysis may not have values that depend directly or indirectly on table fields. A <prompt/description> created for a permanent variable will exist for one session, and must be re-established each time it is used.

Type Indicators

- **TEXTINPUT** - Variables of this data type can receive any alpha-numeric characters including symbols and the output of functions (e.g., FORMAT).
- **NUMERIC** - Variables of this data type can receive numbers and the output of functions (e.g., TXTTONUM).
- **DATEFORMAT** - Variables of this data type can receive date values including the output of functions (e.g., TXTTODATE and NUMTODATE).
- **YN** - Variables of this data type can receive the Boolean values of (+) for Yes and (-) for No. Until an assignment is made, YN type variable values are (.) or missing.

Examples

Example 1: A variable is defined without <scope>, <type indicator>, and <prompt/description>. Standard scope thus becomes the default scope. As no type is specified, the variable can be assigned number, date, text, or Boolean data. However, once assigned a value, the data type of the variable becomes the data type of the value that has been assigned and may not be changed.

```
DEFINE BodyMassIndex NUMERIC
```

Example 2: A variable with standard scope and of type YN is defined.

```
DEFINE DoYouSmoke YN
IF DateSmokingStarted = (+) THEN
   ASSIGN DoYouSmoke = (+)
ELSE
   ASSIGN DoYouSmoke = (.)
END-IF
```

Example 3: A variable with standard scope and in date format is defined.

```
DEFINE DateOfBirth DATEFORMAT
```

Example 4: A variable with permanent scope is defined, but with no <type indicator>. As described in Example 1, the variable data type will be set with the first assignment of data and cannot be changed thereafter.

```
DEFINE StateID PERMANENT
```

Example 5: A variable with global scope and of type text is defined.

```
DEFINE PatientID GLOBAL TEXTINPUT
```

DEFINE DLLOBJECT

Description

This command creates a variable that represents an ActiveX object and a class within an ActiveX DLL (dynamic link library) file, ActiveX EXE (executable file), or a Windows Scripting Component (WSC) file.

Syntax

ActiveX DLL File

```
DEFINE <variable> DLLOBJECT "<ActiveX name>.<class>"
```

Windows Scripting Component (WSC) File

```
DEFINE <variable> DLLOBJECT "<filename>"
```

- The <variable> represents the name of the variable to be created. <variable> cannot be a reserved word.
- The <ActiveX name> represents the internal name of the ActiveX object that contains the class object for the DLL or EXE file.
- The <class> represents an internal class name that is defined within the ActiveX component.
- The <filename> represents the name of the Windows Scripting Component (WSC) file where the script component resides.

Comments

The ActiveX name may not be the same as the actual name of the dll (dynamic link library) or executable (exe) file. An ActiveX object is given a name when it's developed. This name is required to create the object. The ActiveX object (executable or dll) must be registered before it can be used. Windows Scripting Component (WSC) objects can also be used.

Examples

Example 1: A variable is created that points to a class object within the Epiweek DLL file. This variable can subsequently be used to find the Epi Week for any given date.

```
DEFINE Week DLLOBJECT "EIEpiwk.Epiweek"
```

Example 2: A variable is created that points to a class object within the GetGlobalUniqueID.WSC file. You can use this variable to assign a field a unique ID.

```
DEFINE Global_ID DLLOBJECT "GetGlobalUniqueID.WSC"
```

AFTER and END-AFTER

Description

This command divides Check Commands to be executed after data entry. All commands in the AFTER and END-AFTER block are executed *after* entering data to the field, page, or form.

Syntax

```
AFTER
```

- AFTER and END-AFTER are only used in Check Code.

Comments

This command enables actions to occur after accessing a form, page, or field. The default time to execute commands associated with a variable is after entry.

AFTER must start in the first position of the line and end with END-AFTER.

Example

The following commands represent the Check Code for a variable called Demo1.

```
BEFORE
   DIALOG "This is Before entry"
END-BEFORE

AFTER
   DIALOG "This is After entry"
END-AFTER
```

BEFORE and END-BEFORE

Description

This command divides Check Commands to be executed before data entry from those executed after entry. All commands in the BEFORE and END-BEFORE block are executed *before* data entry to the field, page, or form.

Syntax

```
END-BEFORE
```

- ENDBEFORE is only used in Check Code. Command buttons will not take ENDBEFORE since all code inserted will be executed when you click the button.

Comments

This command enables actions to occur before accessing a form, page, or field. The default time to execute commands associated with a variable is before entry.

BEFORE must start in the first position of the line and end with END-BEFORE.

Example

The following commands represent the Check Code that raise a dialog before the cursor enters the field before data entry, and then after the cursor leaves the field following data entry.

```
BEFORE
   DIALOG "This is Before entry"
END-BEFORE

AFTER
   DIALOG "This is After entry"
END-AFTER
```

CLICK and END-CLICK

Description

This command block is for fields that have a click event such as Command Buttons, Checkboxes, Legal Values, and Comment Legal fields. Not all field types support the CLICK command. Check Commands in the Click block are executed immediately upon clicking the field or clicking an item in a drop-down list.

Syntax

```
CLICK
   //Add commands here

END-CLICK
```

Comments

This command block enables actions to occur immediately upon clicking a field that supports the CLICK event.

CLICK must start in the first position of the line and end with END-CLICK.

Example

The following commands raise a dialog when the field is clicked.

```
CLICK
   DIALOG "The checkbox was just clicked."
END-CLICK
```

EXECUTE

Description

Executes a Windows or DOS program - either one explicitly named in the command or one designated within the Windows registry as appropriate for a document with a named file extension. This provides a mechanism for bringing up the default word processor or browser on a computer without first knowing its name. The EXECUTE command accepts a series of paths, separated by semicolons:

```
EXECUTE c:\epi_info\myfile.exe;d:\myfile.exe
```

If the first is not found, the others are tried in succession. In Check Code, the EXECUTE command can be placed in any command block, but is often used with a button. A button does not have a Before Entry section.

Syntax

```
EXECUTE <filename>

EXECUTE "<filename> <command-line parameters>"

EXECUTE NOWAITFOREXIT <filename>

EXECUTE NOWAITFOREXIT '<filename>'

EXECUTE NOWAITFOREXIT "<filename>"

EXECUTE WAITFOREXIT <filename>

EXECUTE WAITFOREXIT '<filename>'

EXECUTE WAITFOREXIT "<filename>"
```

- The <filename> represents the path and program name for .exe (filename for registered Windows programs) and .com (filename for MS-DOS binary executable) files.
- The <command-line parameters> represent any additional command-line arguments that the program can accept.
- When Wait for Command to Execute (modal) is specified, the command should run and Enter should continue running. When Wait for Command to Execute is not specified (non-modal), Enter should wait until the executed program closes before continuing. When EXECUTE is run modally, permanent variables are written before the command is executed and reloaded after it is executed.

Comments

If the name of an executable program, (e.g., ENTER.EXE, MYBATCH.BAT, or MYWEB.HTM) is given, the program will be run in a separate window. The window closes when the program terminates.

If the name given is not a program, but a file with an extension (the three characters after the ".") registered by Windows for displaying the document, the correct program to display the file will be activated (i.e., WRITEUP.DOC might cause Microsoft Word© to run and load the file on one computer). Usually .TXT will run NOTEPAD.EXE© or WORDPAD.EXE©, and image files will appear in a browser or in a graphics program. An .HTM file will bring up the default browser.

You will not need to supply the location or even the name of the program that will be run. These details are stored in the Windows registry for common file extensions. The same concept applies to Internet addresses (URLs). Give a URL (Universal Resource Locator) that ends in .HTM or begins with HTTP:// “http://www.cdc.gov/epiinfo/” and Windows brings up the default browser, connects to the Internet (if possible), and goes directly to the site indicated.

Examples

Example 1: A text file on the C drive is opened. The operating system will select an application to open the file.

```
EXECUTE “C:\logfile.txt”
```

Example 2: An executable file is run.

```
EXECUTE "C:\Windows\Notepad.exe"
```

Example 3: Windows Internet Explorer is run and passed http://www.cdc.gov/epiinfo as a command-line parameter. Because WAITFOREXIT is specified, Enter will not allow you to continue entering data until the browser window is closed.

```
EXECUTE WAITFOREXIT "http://www.cdc.gov/epiinfo"
```

GOTO

Description

This command can be used alone or in an IF statement to transfer the cursor to a named variable field.

Syntax

```
GOTO <event>
```

- The <event> can be a +1, -1, page number, or a field name.

Event and Description

- +1 Automatically saves the current page if changes have been made, and goes to the next page.
- -1 Automatically saves the current page if changes have been made, and goes to the previous page.
- <page number> Automatically saves the current page if changes have been made, and goes to the page indicated by the number.
- <field name> Goes to the field indicated. Automatically saves the current page if changes have been made, if the field is on another page.

Comments

GOTO will be ignored if it is in the Before or After Record event. The GOTO command skips all the variables in between unavailable for data entry or Read Only.

Examples

Example 1: The form will skip directly from one field to another based on certain user input. The example assumes a form exists that has the following fields: DoYouSmoke (Yes/No), PacksPerDay (Numeric), and HeartDisease (Yes/No). The code below would appear in the AFTER section of the DoYouSmoke field.

```
IF DoYouSmoke = (+) THEN
   GOTO HeartDisease
ELSE
   GOTO PacksPerDay
END-IF
```

Example 2: You will be taken to the second page on the form based upon an answer provided to a question on lung disease. The example assumes a form exists that has two pages and the following fields: LungDisease (Yes/No). The code below would appear in the AFTER section of the LungDisease field.

```
IF LungDisease = (-) THEN
   GOTO  2
END-IF
```

Example 3: You will be taken to the page immediately following the one they are on currently. The example assumes a form exists that has two pages and the following fields: DOB (Date) on page 1. The code below would appear in the AFTER section of the DOB field.

```
IF NOT DOB = (.) THEN
   GOTO +1
END-IF
```

UNHIDE

Description

HIDE - This command hides a field from the form and prevents data entry.

UNHIDE - This command makes a field visible and returns it to the status before it was hidden.

Syntax

```
HIDE [<field(s)>]

UNHIDE [<field(s)>]
```

- The <field(s)> represents one or more valid field names.

Comments

If no field name is specified, the current field (the one to which the Check Code block pertains) is assumed. Text fields can be hidden or unhidden to position messages on the screen, and the display of alternate messages. Label/Title fields cannot be hidden.

Examples

Example 1: Questions relating to pregnancy will not be displayed if the patient is male. The ELSE section of the IF command allows the fields to be unhidden if you go back and change your answer in the Sex field. The example assumes a form exists with the following fields: Sex (Text), Pregnant (Yes/No), and ChildBirth (Yes/No). The code below would appear in the AFTER section of the Sex field.

```
IF Sex = "M" THEN
   HIDE Pregnant
   HIDE ChildBirth
ELSE
   UNHIDE Pregnant
   UNHIDE ChildBirth
END-IF
```

Example 2: The field that has the HIDE command will be hidden. The example assumes a form exists with two pages and the following fields: LungDisease (Yes/No). The code below would appear in the AFTER section of the LungDisease field.

```
IF LungDisease = (-) THEN
   HIDE
END-IF
```

Example 3: Questions relating to pregnancy will not be displayed if the patient is a male. The example assumes a form exists with the following fields: Sex (Text), Pregnant (Yes/No), Complications (Yes/No), and ChildBirth (Yes/No). The code below would appear in the AFTER section of the Sex field.

```
IF Sex = "M" THEN
   HIDE Pregnant ChildBirth Complications
```

```
END-IF
```

Note: When hiding a field, it is important that be unhidden (with an If…Then…Else…) if the value entered is changed.

IF THEN ELSE

Description

This command defines conditions and one or more consequences which occur when the conditions are met. An alternative consequence can be given after the ELSE statement to be realized if the first set of conditions is not true. The ELSE statement is optional.

Syntax

```
IF <expression> THEN

   [command(s)]

END-IF


IF <expression> THEN

   [command(s)]

ELSE

   [command(s)]

END-IF
```

- The <expression> represents a condition that determines whether or not subsequent commands will be run. If the condition evaluates to true, the commands inside of the IF block will run. If the condition evaluates to false, the commands inside of the ELSE block will run instead. If no ELSE exists and the condition is false, then no commands inside of the IF block are run.
- The (command[s]) represents at least one valid command.
- The ELSE statement is optional and will run any code contained inside of it when the <expression> evaluates to false.

Comments

The IF statement is executed immediately if it does not refer to a database variable, any characteristic or attribute that can be measured, or if any defined variables have been assigned literal values.

Examples

Example 1: If you select "Male" for the patient's sex then the fields named Pregnancy and ChildBirth are hidden. The example assumes a form exists with the following fields all on the same page: Sex (Text), Pregnancy (Yes/No), and ChildBirth (Yes/No). The code below would appear in the AFTER section of the Sex field.

```
IF (Sex = "Male") THEN

   HIDE Pregnancy Childbirth
```

```
END-IF
```

Example 2: If the date of birth supplied by the user occurs prior to January 1, 1900, the Enter module provides a warning beep and a warning dialog indicating invalid input. However, if the date of birth supplied is on or after January 1, 1900, the GOTO command is executed instead taking you to the following page. The example assumes a form exists with the following fields: DOB (Date). It also assumes a page exists following the page where DOB resides. The code below would appear in the AFTER section of the DOB field.

```
IF DOB < 01/01/1900 THEN

   BEEP

   DIALOG "Warning: Invalid date of birth detected"

ELSE

   GOTO +1

END-IF
```

Example 3: The date of birth field is validated to ensure correct input. The date of birth field must not be less than January 1, 1900, must not be greater than the current time, and must not be greater than the date of the survey. If any of these conditions is not met, a warning dialog is displayed and the invalid input erased. The example assumes a form exists with that has the following fields: DOB (Date) and SurveyDate (Date). The code below would appear in the AFTER section of either DOB or SurveyDate, depending on whichever one will be filled in last.

```
IF (DOB < 01/01/1900) OR (DOB > SYSTEMDATE) OR (DOB > SurveyDate) THEN

   BEEP

   DIALOG "Warning: Invalid date of birth detected"

   CLEAR DOB

END-IF
```

Example 4: An IF command is used to check if a field has been left blank. The example assumes a form exists with the following field: LastName (Text). The code below would appear in the AFTER section of the LastName field.

```
IF NOT LastName = (.) THEN

   BEEP

   DIALOG "Last name field should not be blank."

END-IF
```

Example 5: Multiple IF commands are used to generate more than two possible outcomes. The example assumes a form exists with the following fields: AgeType (Text), AgeYears

(Numeric), and Age (Numeric). The code below would appear in the AFTER section of AgeType or Age, depending on which one will be filled in last.

```
IF AgeType = "Days" THEN

   ASSIGN AgeYears = Age / 365.25

END-IF

IF AgeType = "Months" THEN

   ASSIGN AgeYears = Age / 12

END-IF

IF AgeType = "Years" THEN

   ASSIGN AgeYears = Age

END-IF
```

Example 6: The AND operator requires both Sex to be "F" and Pregnancy to be true in order for the GOTO command to be executed. The example assumes a form exists with the following fields: Sex (Text), Pregnancy (Yes/No), and ChildBirth (Yes/No). The code below would appear in the AFTER section of either Sex or Pregnancy, depending on which one is filled in last.

```
IF (Sex = "F") AND (Pregnancy = (+)) THEN

   GOTO ChildBirth

END-IF
```

Example 7: Several IF commands are used to determine if a patient is ill. If any one of the symptoms listed in the form are true, the field ill is assigned true. The example assumes a form exists that has the following fields: Ill (Yes/No), Vomiting (Yes/No), Fever (Yes/No), and Diarrhea (Yes/No). The code below would appear in the AFTER section of the ill field.

```
ASSIGN Ill = (-)
IF Vomiting = (+) THEN

   ASSIGN Ill = (+)

END-IF

IF Diarrhea = (+) THEN

   ASSIGN Ill = (+)

END-IF

IF Fever = (+) THEN

   ASSIGN Ill = (+)

END-IF
```

Example 8: Several IF commands are used to determine the number of symptoms a patient is presenting with. If the number of symptoms is greater than or equal to two, the Case variable is assigned true. If the number of symptoms is less than two, the Case variable is assigned false. The example assumes a form exists that has the following fields: MajorSymp (Numeric), Vomiting (Yes/No), Fever (Yes/No), Diarrhea (Yes/No), and Case (Yes/No).

```
ASSIGN MajorSymp = 0
IF Diarrhea = (+) THEN

   ASSIGN MajorSymp = MajorSymp + 1

END-IF

IF Fever = (+) THEN

   ASSIGN MajorSymp = MajorSymp + 1

END-IF

IF Vomiting = (+) THEN

   ASSIGN MajorSymp = MajorSymp + 1

END-IF

IF MajorSymp >= 2 THEN

   ASSIGN Case = (+)

ELSE

   ASSIGN Case = (-)

END-IF
```

NEWRECORD

Description

This command saves the current records data and opens a new record for data entry.

Syntax

```
NEWRECORD
Example

DIALOG "This is the last field in my form."
NEWRECORD
```

Analysis Commands

ASSIGN

Description

This command assigns numeric or string expression results to a variable. It may be a database variable in a form or data table, or a user-defined variable created by the DEFINE command in a program.

Syntax

```
ASSIGN <variable> = <expression>

LET <variable> = <expression>
```

(ASSIGN and LET may be omitted)

<variable> = <expression>

- The <variable> represents a variable in a database or a defined variable created in a program.
- The <expression> represents any valid arithmetic or string expression.

Program Specific Feature

If the right side of the assignment does not contain a field variable (one in a database table), or a variable that depends on a field variable, the assignment is made immediately.

```
DEFINE YEAR NUMERIC

ASSIGN YEAR = 2000
```

The following code contains two view variables, ONSETDATE and EXPOSUREDATE.

```
DEFINE INCUBATION NUMERIC

ASSIGN INCUBATION = ONSETDATE-EXPOSUREDATE
```

In this example, INCUBATION is only calculated during current dataset processing. It is calculated for each record and may be used similar to a dataset variable in procedures (i.e., TABLES, FREQ, and GRAPH). Prior to and after processing a dataset, INCUBATION will have a "missing" value, although it could be assigned a value with another statement (e.g., INCUBATION = 999).

The value is calculated each time a record that meets the conditions of SELECT is read from the dataset. Any legal expression can be used that combines functions or literal values and operators (i.e., &, +, -, *, /, ^, and MOD). Boolean expressions are not supported in assign commands. Standard variables that depend on database fields must be saved to a table using WRITE before they can be edited using LIST UPDATE.

Comments

Temporary variables must be defined before being used and will accept any type of data (i.e., text, numeric, or date). Once they have been assigned a non-missing value or an expression, their type cannot change.

If an attempt is made to assign an invalid expression to a variable, it retains any previous assignment.

Examples

Example 1: The ASSIGN command is used to assign values to defined variables and database variables. Note that literal values (e.g., 42, other variables, and functions) can be used on the right side of the = operator.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Surveillance

DEFINE State TEXTINPUT
ASSIGN City = "Atlanta"
ASSIGN State = "GA"
ASSIGN Address = City & ", " & State
DEFINE Now DATEFORMAT
ASSIGN Now = SYSTEMTIME
DEFINE Duration NUMERIC
ASSIGN Duration = YEARS(01/01/1998, ReportDate)
DEFINE Ill YN
ASSIGN Ill = (-)
ASSIGN Age = 42
ASSIGN Occupation = "Doctor"
LIST Address City State Duration Now Ill Occupation Age GRIDTABLE
```

BEEP

Description

This command generates a sound.

Syntax

```
BEEP
```

Example

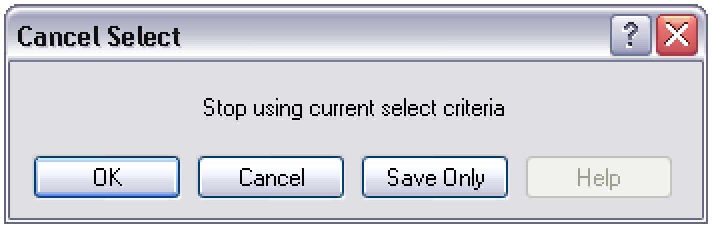
If the number of records in the database is greater than 1,000, a beep is generated and a dialog box appears.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Surveillance

IF RECORDCOUNT > 4 THEN

   BEEP

   DIALOG "Database greater than 4 records."

END
```

CANCEL SELECT or SORT

Description

This command cancels a previous SELECT or SORT command.



Syntax

```
SORT

SELECT
```

Comments

The Cancel Select and Cancel Sort commands automatically close the current output file.

Example

The commands below should be run one-by-one to better understand how the cancel sort and cancel select commands function.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego

SELECT Ill = (+)
LIST Age Ill Sex
SELECT
SORT Age
LIST Age Ill Sex
SORT
```

CLOSEOUT

Description

This command closes the current output file. It is normally used after a ROUTEOUT command when all the information to be included in the ROUTEOUT file has been produced.

Syntax

```
CLOSEOUT
```

Example

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego

ROUTEOUT “C:\ My_Project_Folder \Outbreak1.htm” REPLACE
TABLES Vanilla Ill STRATAVAR=Sex
MEANS Age Ill
CLOSEOUT
```

COXPH

Description

This command performs Cox-Proportional Hazards and Extended Cox-Proportional Hazards survival analysis. This form of survival analysis relates covariates to failure through hazard ratios. A covariate with a hazard ratio greater than one causes failure. A covariate with a hazard ratio less than one improves survival. Some of the subjects may be unavailable prior to failure; the term "censored" is applied to them. COXPH is especially constructed to deal with this situation. Statistics showing the risk set by group and time can be written to an OUTTABLE for later formatting.

Syntax

```
COXPH <time variable>= <covariate(s)>[: <time function>:]  *  <censor variable>
(<value>) [TIMEUNIT="<time unit>"] [OUTTABLE=<tablename>] [GRAPHTYPE="<graph type>"]
[WEIGHTVAR=<weight variable>] [STRATAVAR=<strata variable(s)>] [GRAPH=<graph
variable(s)>]
```

- The <time variable> represents a numeric or date variable, specifying when failure or censorship occurred.
- The <covariate(s)> represent a numeric variable, a non-numeric variable, or a variable specified as non-numeric by parenthesis. Any non-numeric variable, even a variable specified as non-numeric by surrounding with parenthesis, is automatically recoded into dummy variables. For all but one of the levels of a variable, a dummy variable will be created. It measures the contribution of its level to the excluded level. A covariate may be followed by a time function. This causes COXPH to run the Extended Cox procedure.
- The <time function> represents a numeric expression involving the time variable.
- The <censor variable> indicates whether the event is a failure or a censor.
- The <value> indicates which value of the CensorVar represents failure.
- The <strata variable(s)> represents a list of variables indicating the different levels of strata.
- The <weight variable> represents a variable to specify the contribution each data row has on the output.
- The <time unit> represents a value for labeling the time axis.
- The <tablename> represents a valid table name.

The <graph type> generates one of the indicated graphs:

1. Survival Probability shows the adjusted survival curves.
2. Observed shows the observed survival curves.
3. Survival-Observed shows the adjusted and observed survival curves.
4. Log-log Survival shows the logarithm of the negative of the logarithm of the adjusted survival curve.
5. Log-log Observed shows the logarithm of the negative of the logarithm of the observed survival curve.
6. Hazard Function shows the adjusted hazard function.
7. None

- The <graph variable(s)> represent a list of variables used to generate survival curves. Graph variables that are covariates or strata variables create curves adjusted by the covariates at all possible combinations of these graph variables. If a variable is numeric, it is plotted at its average value. Otherwise the graph variable splits the data into separate groups, each with its own curve.

Comments

COXPH uses the Breslow method to handle ties in the data.

Example

In this example, we will use the Anderson dataset from a clinical trial of leukemia patients to compare the treatment and placebo group survival.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Anderson
COXPH STIME = (Rx) * Status ( 1 ) TIMEUNIT="Weeks" PVALUE=95% GRAPH=Rx
GRAPHTYPE="Survival Probability"
```

DEFINE

Description

This command creates a new variable.

Syntax

```
DEFINE <variable> (<scope>) (<type indicator>)
```

- <variable> represents the name of the variable to be created.
- <scope> is the level of visibility and availability of the new variable and must be one of the reserved words STANDARD, GLOBAL, or PERMANENT. If omitted, STANDARD is assumed and a type indicator may not be used.
- <type indicator> **is required** and cannot be used if <scope> is omitted. <type indicator> is the data type of the new variable and must be one of the following reserved words: NUMERIC, TEXTINPUT, YN, or DATEFORMAT. If omitted, the variable type will be inferred based on the data type of the first value assigned to the variable. However, omitting field type **is not** recommended. Once field type is defined, the variable type cannot be changed. An error will occur if you attempt to assign data of a different type to the variable.

Comments

A custom variable defined in Epi Info 7 might not have a predefined data type if the <type indicator> is omitted when the variable is defined. If the variable does not have a predefined type and has not been used, the variable will accept a value in any of the four data types (Text, Number, Date, Yes/No [Boolean]) that is assigned to the variable the first time. Thereafter, the variable takes on the data type of the value assigned and it's data type cannot be changed. However, omitting field type is not recommended. An error will occur if you attempt to assign data of a different data type. Various functions can be used to manipulate data, changing data type of values to match the data type of the variable. Some of these functions include FORMAT, TXTTONUM, TXTTODATE, and NUMTODATE.

Variable Scope

- **STANDARD** variables retain their value only within the current record and are reset when a new record is loaded. Standard variables are used temporarily behaving like other fields in the database. In Classic Analysis, Standard variables lose their values and definitions with each READ statement.

- **GLOBAL** variables retain values across related forms and when the program opens a new form, but are removed when the Classic Analysis program is closed. Global variables persist during program execution. Global variables are also used to store values between changes of data source (e.g., when the READ command is used). Global variables in Classic Analysis may not depend directly or indirectly on table fields.
- **PERMANENT** variables are stored in the EpiInfo.Config.xml file and retain any value assigned until the value is changed by another assignment or the variable is undefined. Permanent variables are shared among Epi Info 7 programs (i.e., Menu, Enter, Classic Analysis, etc.) and persist even if the computer shuts down. Permanent variables in Classic Analysis may not have values that depend directly or indirectly on table fields. A <prompt/description> created for a permanent variable will exist for one session and must be re-established each time it is used.

Type Indicators

- **TEXTINPUT** - Variables of this data type can receive any alpha-numeric characters including symbols and the output of functions (e.g., FORMAT).
- **NUMERIC** - Variables of this data type can receive numbers and the output of functions (e.g., TXTTONUM).
- **DATEFORMAT** - Variables of this data type can receive date values including the output of functions (e.g., TXTTODATE and NUMTODATE).
- **YN** - Variables of this data type can receive the Boolean values of (+) for Yes and (-) for No. Until an assignment is made, YN type variable values are (.) or missing.

Example

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Surveillance

DEFINE Birthday DATEFORMAT
ASSIGN Birthday = 01/01/2006
DEFINE HospitalCode NUMERIC
ASSIGN HospitalCode = 854
DEFINE Smoke YN
ASSIGN Smoke = (+)
LIST Birthday HospitalCode Smoke
```

DEFINE DLLOBJECT

Description

This command allows you to create an ActiveX (dynamic link library or executable) object.

Syntax

```
DLL File

DEFINE <variable> DLLOBJECT "<ActiveX name>.<class>"
WSC File

DEFINE <variable> DLLOBJECT "<filename>"
```

- The <variable> represents the name of the variable to be created.
- The <ActiveX name> represents the internal name of the ActiveX object that contains the class object for DLL files.
- The <class> represents a class name defined within the DLL.
- The <filename> represents the name of the WSC file where the script component resides.

Comments

The ActiveX name may not be the same as the actual name of the dll (dynamic link library) or executable. When an ActiveX object is developed, it is given a project name. This name is required to create the object. The ActiveX object (executable or dll) must be registered before it can be used. Windows Scripting Component (WSC) objects can also be used.

Example

```
DEFINE Week DLLOBJECT "EIEpiwk.Epiweek"
```

Define Group Command (Analysis Reference)

Description

This command allows you to create temporary group variables

Syntax

```
DEFINE <variable> GROUPVAR <variable 1> [<variable 1> ...]
```

- <variable> represents the new group variable being defined.
- <variable 1> ... represent the existing variables to which the new group variable will be equivalent.

Program Specific Feature

Field variables, not defined variables must be in the group. Variables in the group may be from different pages of the same or different forms, in any combination. They may themselves be group variables, but each variable will be represented in the new group only once no matter how many times it appears in the variable list or in group variables in the variable list.

If the group variable being defined exists and is not a group variable, an error message is displayed and the command is not processed. If it is a group variable, the new variable list replaces the existing variable list until the next READ or MERGE command or until it is redefined again. Group variables cannot be undefined.

Comments

This command is useful when there are many variables so that the use of * is impractical or invalid. Group variables can be used in the LIST and WRITE commands and in some statistical commands (e.g., FREQ, TABLES, and MEANS). Group variables cannot be used in Complex Sample commands.

Example

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego

DEFINE Exposure GROUPVAR JELLO CAKES VANILLA CHOCOLATE
LIST Exposure
```

DELETE FILE/TABLES

Description

This command deletes files, tables, and forms.

Syntax

```
DELETE filename | wildcard {RUNSILENT}
```

```
DELETE TABLES tablename | filename:tablename {RUNSILENT | SAVEDATA}
```

Comments

Delete file specifies explicitly or implicitly (via wildcards) files to be deleted. If no files are specified, or if some of the specified files cannot be deleted, a message is produced unless RUNSILENT is specified. Wildcards are not permitted in the suffix.

Delete table specifies a table to be deleted. If the table does not exist or cannot be deleted, a message is produced unless RUNSILENT is specified. Unless RUNSILENT is specified, a confirmatory message is displayed prior to deletion.

To delete tables with spaces in their names, specify both the file and the table even if the table is in the current project. The table must be enclosed in single quotes. You can read a table with a space in its name, delete it, causing errors during subsequent procedures. To delete a table with space in its name, specify the database even if the table is in the current database.

DELETE TABLES will not delete, and produce a message if any of the specified tables are data, grid, or program. Code tables are deleted only if they are not referenced by any view.

SAVEDATA does not delete the data tables, but removes the RECStatus property from the table.

Space is not reclaimed until the Compact utility is run.

Analysis will not delete the current project MDB or any table in use by the current form.

Example

```
DELETE TABLES Backup2005
```

DELETE RECORDS

Description

This command deletes selected records.

Syntax

```
DELETE * {PERMANENT | RUNSILENT}
```

```
DELETE expression {PERMANENT | RUNSILENT}
```

Comments

All records in the current selection are set to deleted status, or if PERMANENT is specified, physically deleted. A confirmation message is displayed unless RUNSILENT is selected. This applies to Access tables and may not be used when using related tables. Use Epi Info 3.5.3 Compact Database utility to reclaim space after deleting tables.

Examples

Example 1: The code below will permanently erase all records in the current data source where the AgeInDays variable contains a value greater than five. Permanent deletions cannot be undone.

```
DELETE AgeInDays > 5 PERMANENT
```

Example 2: The code below will mark all records in an Epi Info 7 database for deletion. The UNDELETE Command can restore records that have been marked for deletion.

```
DELETE *
```

DIALOG

How to Use the DIALOG Command

Description

This command provides interaction with the user from within a program. Dialogs can display information, ask for and receive input, and offer lists for making choices.

Syntax

Simple Form

```
DIALOG "<prompt>" {TITLETEXT="<title>"}
```

- <prompt> represents the text to be displayed as message.
- <title> represents the text to be used as the caption for the dialog window. If <title> is omitted in the Dialog command, "Analysis" will be displayed on the dialog box's title bar.

Get Variable Form

```
DIALOG "<prompt>" <variable> <entry type> {TITLETEXT="<title>"}

DIALOG "<prompt>" <variable> "<value 1>", "<value 2>", "<value 3>", … ,"<value n>"
{TITLETEXT="<title>"}

DIALOG "<file type selection>" <variable> READ {TITLETEXT="<title>"}

DIALOG "<file type selection>" <variable> WRITE {TITLETEXT="<title>"}
```

- <variable> represents the variable to store value entered.
- <entry type> represents a reserved word and/or mask that defines the type of input to be accepted and stored in the variable. The following are valid entry types:
    - ➢ **TEXTINPUT** specifies the input as text.
    - ➢ **YN** specifies the input as Boolean.
    - ➢ **DATEFORMAT** ("<date mask>") specifies the input as a date; if a date mask is not specified, the system short date is used.
    - ➢ "<numeric mask>" specifies the input as numeric. This option is identified as "Number Only". The numeric mask should be a text string (e.g., "####" or "##.###") that indicates the type of data to be entered. # indicates a digit.
    - ➢ If no <entry type> is specified, the input variable is interpreted as a number if possible. If the value is not a valid number, but is a valid date, it is interpreted as a date. Otherwise, an error occurs.
- <value> represents a value in a drop-down list of choices. Each value included in the command will be shown as a single item in the list.
- <file type selection> controls the type of files that are displayed for selection. It consists of alternating description and filter elements separated by vertical bars. The description is displayed for you to select. The corresponding filter element selects the files. If this element is left blank, all files can be selected. Syntax is created using the File Open and File Save options from the Dialog Format drop down menu.

List of Values Form

```
DIALOG "<prompt>" <variable> [<list type>] {TITLETEXT="<title>"}

DIALOG "<prompt>" <variable> DBVALUES <table name> {TITLETEXT="<title>"}
```

- The [<list type>] are DATABASES and DBVIEWS

Comments

The Simple form of DIALOG places a dialog box on the screen, using the text provided, with an OK button.

The Get Variable form of DIALOG displays the text prompt and provides a means for entering a value.

- If **YN** is specified, "Yes," "No" and "Cancel" buttons are presented and the specified variable is set to (+), (-), or (.).
- If **TEXTINPUT** or no entry type is specified, an entry field for user response is provided with OK and Cancel buttons. The variable is assigned the value of the input with a missing value if Cancel is chosen.
- The **Get Variable** form of the DIALOG command can also display a file selection dialog. If READ is used, only existing files are displayed. If WRITE is used and an existing file is selected, you will see the Overwrite? prompt.
- The **Multiple Choice** form of DIALOG displays the text prompt and provides a combo box for selecting among the values with OK and Cancel buttons. The variable is assigned the value of the input with a missing value if Cancel is chosen. Variable will text type.
- The **Date** form of the DIALOG command uses a special control for accepting input. Each section of the date (year, month, and day) can be increased or decreased independently using the drop down calendar. Using this control, it is impossible to select an invalid date.

The List of Values form of DIALOG displays a combo box of databases or form variables with OK and Cancel presented. The variable is assigned the value of your input with a missing value if Cancel is chosen. Variable will be text type. The VARIABLE VALUE form of DIALOG displays a combo box of distinct values of the specified variable in the specified database. The variable is assigned the value of input with a missing value if Cancel is chosen. The variable will be of the same type as the database variable.

Date Mask	
*	System Time Format
!	System Long Date

Numeric Mask	
#	Digit placeholder (Entry required).
.	Decimal placeholder. The actual character used is the one specified as the decimal placeholder in the computer's international settings.
,	Thousand separator. The actual character used is the one specified as the thousands placeholder in the computer's international settings.
9	Digit placeholder. (Entry optional).
Punctuation	Included in the display.

Custom Formats	
D	One- or two-digit day.
Dd	Two-digit day. Single-digit day values are preceded by a zero.
Ddd	Three-character weekday abbreviation.
dddd	Full weekday name.
H	One- or two-digit in a 12- hour format.
Hh	Two-digit hour in a 12-hour format. Single digit values are preceded by a zero.
H	One- or two-digit hour in 24-hour format.
HH	Two-digit hour in a 24-hour format. Single digit values are preceded by a zero.
M	One- or two-digit minute.
Mm	Two-digit minute. Single digit values are preceded by a zero.
M	One- or two-digit month number.
MM	Two-digit month number. Single digit values are preceded by a zero.
MMM	Three-character month abbreviation.
MMMM	Full month name.

Custom Formats	
S	One- or two-digit seconds.
Ss	Two-digit seconds. Single- digit day values are preceded by a zero.
T	One-letter AM/PM abbreviation. AM displays as A.
Tt	Two-letter AM/PM abbreviation. AM displays as AM.
Y	One-digit year. 1997 displays as 7.
Yy	Last two digits of the year. 1997 displays as 97.
Yyyy	Full year. 1997 displays as 1997.
Punctuation	Included in the display

Examples

Example 1: A prompt is displayed letting you know that the following commands may take several minutes to complete because of their complexity.

```
DIALOG "Warning: This script may take several minutes to complete" TITLETEXT="Warning"
```

Example 2: The DIALOG command is used to obtain a user-supplied date to calculate the patient's age. If you do not supply a date (press the Cancel button), the default ending date contained in the survey data source is used instead.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Surveillance
DEFINE UpdateDate DATEFORMAT
DEFINE SubmitDate Dateformat
ASSIGN SubmitDate = EventDate
DIALOG "Enter a new ending date for survey data:" UpdateDate DATEFORMAT "MM-DD-YYYY"
DEFINE NewAge NUMERIC
ASSIGN NewAge = YEARS(SubmitDate, UpdateDate)
GRAPH NewAge GRAPHTYPE="Column"
```

Example 3: You will see a drop-down list of choices. The list contains part of the command; in this case, several counties in the state of Georgia. Your selection is temporarily assigned to all records in the form.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego

DEFINE NewCounty TEXTINPUT
DIALOG "Select a county" NewCounty "Fulton", "Baldwin", "DeKalb", "Cobb"
ASSIGN CountyName = NewCounty

LIST CountyName
```

Example 4: A drop-down list of choices is displayed. Each list item is retrieved from the X_COORD column of the SohoDead data table.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}::SohoDead
DEFINE Coordinates TEXTINPUT
DIALOG "Select Coordinates" Coordinates DBVALUES SohoDead X_COORD
TITLETEXT="Coordinates"
```

Example 5: A dialog box prompts you for an image file in JPG or bitmap format. Upon choosing the file, the full path and filename of the file are saved to the specified variable.

```
DEFINE FileName TEXTINPUT
DIALOG "Image files|*.bmp;*.jpg;|Allfiles|*.*" FileName READ TITLETEXT="Get image
files"
```

Example 6: A dialog box is displayed that allows you to enter a number. An input mask is used to force your input to three whole digits and one decimal digit. Your input is temporarily assigned to each record in the form.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego

DEFINE NewAge NUMERIC
DIALOG "Enter patient age" NewAge "###.#"
ASSIGN Age = NewAge
LIST Age
```

Example 7: A dialog box is displayed that allows you to enter a number. An input mask is used in order to force your input to either one or two whole digits and one decimal digit. (The 9 represents an optional digit.) Your input is temporarily assigned to each record in the form.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego

DEFINE NewAge NUMERIC
DIALOG "Enter patient age" NewAge "9##.#"
ASSIGN Age = NewAge
LIST Age
```

DISPLAY

Command Generator

How to Use the DISPLAY Command

Description

This command displays table, form, and database information.

Syntax

```
DISPLAY <option> [<sub-option>] [OUTTABLE=<tablename>]
```

- The <option> determines the type and source of information displayed.
- The <tablename> represents the name of the data output table to receive results (optional).
- The [<sub-option>] refers to the type of displayed information.
    1. **Option DBVARIABLES** - displays information about variables. Sub-option DEFINE displays only defined variables. Sub-option FIELDVAR displays only table/view variables for the current READ and RELATE tables. Sub-option LIST followed by a list of variable names displays only the specified variables.
    2. **Option DBVIEWS** - displays information about forms and other Epi Info 7 specific tables in a database. The sub-option is the path of the database. Omitting the sub-option displays information for the current project database.
    3. **Option TABLES** - displays information about tables in a database, whether Epi Info 7 specific or generic. The sub-option is the path of the database. Omitting the sub-option displays information for the current project database.

Examples

Example 1: The DISPLAY command is used to show the tables, forms, and variables from an existing data source.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego

DISPLAY TABLES
DISPLAY DBVIEWS
DISPLAY DBVARIABLES
```

Example 2: All forms in the Sample database are displayed.

```
DISPLAY DBVIEWS 'C:\Epi_Info_7\Projects\Sample\Sample.prj’
```

Example 3: All tables in the Sample database are displayed.

```
DISPLAY TABLES 'C:\Epi_Info_7\Projects\Sample\Sample.prj’
```

Example 4: All variables in Oswego are written to a new table in the Sample database called VarInfo.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego

DISPLAY DBVARIABLES OUTTABLE=VarInfo
```

EXECUTE

Description

This command executes a Windows program; one explicitly named in the command, or one designated within the Windows registry as appropriate for a document with a named file extension. This provides a mechanism for bringing up whatever word processor or browser is the default on a computer without first knowing its name.

If the pathname is a long filename, it must be surrounded in single quotes. If the command takes parameters, surround the command and the parameters with a single set of double quotes. Do not use single quotes.

Program Specific Feature

Link and Activate are not valid command names.

Syntax

```
EXECUTE <filename>

EXECUTE "<filename> <command-line parameters>"

EXECUTE NOWAITFOREXIT <filename>

EXECUTE NOWAITFOREXIT '<filename>'

EXECUTE NOWAITFOREXIT "<filename>"

EXECUTE WAITFOREXIT <filename>

EXECUTE WAITFOREXIT '<filename>'

EXECUTE WAITFOREXIT "<filename>"
```

- The <filename> represents the path and program name for .exe (filename for registered Windows programs) and .com (filename for MS-DOS binary executable) files.
- The <command-line parameters> represent any additional command-line arguments that the program can accept. When used, the entire string should be enclosed within double quotes.
- When Wait for Command to Execute (modal) is specified, the command and Analysis should run. When Wait for Command to Execute is not specified (non-modal), Classic Analysis should wait until the executed program closes before continuing. When EXECUTE is run modally, permanent variables are written before the command is executed, and reloaded after the command is executed.

Comments

If the name given is not a program, but a file with an extension (the three characters after the ".") registered by Windows for displaying the document, the correct program to display the file will be activated (i.e., WRITEUP.DOC might cause Microsoft Word © to run and load the file on one computer). On another machine, however, .DOC might correspond to Corel WordPerfect©. Usually .TXT will run NOTEPAD.EXE© or WORDPAD.EXE©, and image files will appear in a browser or in a graphics program. An .HTM file will bring up the default browser.

Examples

Example 1: The EXECUTE command is used to start the Enter module. The command-line parameter is used to load the Oswego Form from the Sample database.

```
EXECUTE "C:\Epi Info 7\

Epi Info 7\Enter.exe"
```

Example 2: The output file generated (in this case, Outbreak1.htm) is opened in the computer's default web browser.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego

ROUTEOUT "C:\Epi Info 7\Epi Info 7\Output\Outbreak1.htm' REPLACE
SET STATISTICS=COMPLETE
TABLES Vanilla Ill STRATAVAR=Sex
MEANS Chocolate Ill
CLOSEOUT
EXECUTE "C:\Epi Info 7\Epi Info 7\Outbreak1.htm"
```

FREQ

Description

FREQ produces a table from the table(s) specified in the last READ statement, showing how many records have each value of the variable. Exact Confidence limits for each proportion are included.

Syntax

```
FREQ [<variable(s)>] {<settings>}

FREQ * {EXCEPT [<variable(s)>]} {<settings>}
```

- <variable(s)> represents one or more variable names. Group variables may be used.
- <settings> represent clauses from the SET command indicating a value of a setting (except PROCESS and HYPERLINKS) which will be used for the duration of the statistical command only.

Comments

Records may be included or excluded from the count by using SELECT statements. Those marked as deleted in Enter will be handled according to the current setting for SET PROCESS. If more than one variable name is given, FREQ makes a separate table for each variable. Confidence limits for the binomial proportions are produced.

If a WEIGHTVAR is specified, the value of the WEIGHTVAR variable is treated as a count of instances of the variable being computed in the frequency (i.e., in the following command a record containing the value 30 for AGE and 15 for COUNT would give a result equivalent to 15 individuals of age 30).

FREQ AGE WEIGHTVAR = COUNT

If STRATUM is specified, a separate frequency is produced for each stratifying variable value.

FREQ ILL STRATAVAR=SEX, produces a table showing ILL (Yes/No/Unknown) for males and another for females. The same numbers can be obtained using TABLES ILL SEX, but the latter produces results in one table rather than in separate tables, and produces statistics to test for an association between ILL and SEX.

FREQ * makes a table for each variable in the current form other than unique identifiers. It is often used to begin analyses of a new data set.

To do frequencies of all variables except a few, use FREQ * EXCEPT VarName(s) followed by the names of the variables to be excluded.

Multiline (memo) variables cannot be used in Frequencies. To use a Multiline variable, define a new variable and assign to it the value SUBSTRING(<old variable>,1,255) and use it in the frequency.

Examples

Example 1: The number of ill and healthy people are displayed along with their percentages and the total.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
FREQ Ill
```

Example 2: In this case, the variable 'Desserts' is a group variable containing the Yes/No variables Chocolate, Vanilla, and Cakes. Running a frequency on a group variable automatically runs a frequency on every variable contained in the group.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
FREQ Desserts
```

Example 3: A frequency on two variables is produced. In this case, BakedHam and Milk.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
FREQ BakedHam Milk
```

Example 4: A frequency on every variable in the current data source is produced.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
FREQ *
```

Example 5: A frequency on every variable in the current data source (except Age, Code_RW, DateOnset, Name and TimeSupper) is produced.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
FREQ * EXCEPT Age Code_RW DateOnset Name TimeSupper
```

Example 6: A frequency of ill people is produced, stratified by sex. Using the stratification option will produce two frequencies. In this case, males and females.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
FREQ Ill STRATAVAR=Sex
```

Example 7: A weighted frequency is conducted. For each record, the value stored in Count is used to represent that record's weight.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Lasum
FREQ Outcome WEIGHTVAR=Count
```

Example 8: A complex sample frequency is run using the Epi1 data set.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Epi1
SET STATISTICS=COMPLETE
FREQ VAC PSUVAR=Cluster
```

GRAPH

Description

This command produces graphs from data in Classic Analysis.

Syntax

```
GRAPH [<variable(s)>] GRAPHTYPE="<graph type>" [<option name>=option value]

GRAPH [<variable(s)>] * <crosstab> GRAPHTYPE="<graph type>"

GRAPH [<variable(s)>] GRAPHTYPE="<graph type>" XTITLE="<string>" YTITLE="<string>"
```

- The <variable(s)> represents the (main) variable(s) to be graphed.
- The <crosstab> represents a variable to be used to classify the main variable(s).
- The <graph type> represents one of the graph types described below.
- **STRATAVAR=<variable>** generates a graph for each value of VarName. When saving a template, the template stores the current graph's background including the main title and the subtitle if it exists. However, if the sub-title is stratified (under "One Graph for Each Value of" in the dialog screen), it will be filtered out.
- **WEIGHTVAR=<variable>** weights each record by the value of <VarName>.
- **WEIGHTVAR=<agg func>(<variable>)** allows multiple records referring to the same values of the main variable, crosstab variable (if any), and strata variable (if any) are represented by the aggregate of VarName. Permissible AggFunc values are MIN (minimum), MAX (maximum), AVG (average), STDEV (standard deviation), SUM, SUMPCT (percent based on total of VarName), COUNT (number of records), or PERCENT (percent based on number of records).
- **TITLETEXT="<string>"** represents a title for each page of graphs. This title is in addition to the title for each graph, which is set by customization.
- **DATEFORMAT="<format string>"** is used when a main variable is a date variable to determine the format in which it will be displayed. Uses the same coding scheme as the FORMAT function.
- **XTITLE="<string>", YTITLE="<string>"** are used to pass X- and Y-axis labels from the GRAPH command.

1.1.1.1 Program Specific Feature

Multiline (memo) fields cannot be graphed. To use a Multiline variable, define a new variable, assign to it the value SUBSTRING(<old variable>,1,255), and use it in the graph.

The following are the graph types capable of being generated by the GRAPH command along with type-specific information ("series" refers to the values of a main variable for a specific value of any crosstab and strata variables):

- **Line graphs** connect X-Y points with lines. The main variables are the X and Y variables. Each series is represented by a different style of line. Both variables must be numeric. To generate a line graph with categorical data, generate a Bar graph and use customization.
- **Column graphs** use vertical bars to represent the count or weight for each value of the main variable(s). Each series results in an additional vertical bar at each point; the bars are distinguished by their style.
- **Bar graphs** use horizontal bars to represent the count or weight of each value of the main variable(s). Each series creates an additional horizontal bar at each point.
- **Epi Curve** use vertical bars to represent the count or weight for each value of the main variable. Each series creates an additional vertical bar at each point. The main variable must be numeric. This graph differs from the bar graph because adjacent bars represent equal ranges of the main variable.
- **Pie** in which each series is represented by a circle, and each value of the main variable has a slice of the circle proportional to the value associated with it.
- **Area** is similar to a line, except that the area below the line contains a solid fill.
- **Scatter graphs** display X-Y points as a scatter gram. The main variables are X and Y variables. Each series is represented by a different style of point.
- **Bubble graphs** are a variation of a Scatter chart in which the data points are replaced with bubbles. A Bubble chart can be used instead of a Scatter chart if your data has three data series.

Examples

Example 1: A graph showing the age of all survey respondents is displayed.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
GRAPH Age GraphType="Column"
```

Example 2: A pie chart showing age categories is displayed.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
DEFINE AgeCategory TEXTINPUT

RECODE Age TO AgeCategory

LOVALUE - 0 = "Unknown"

0 - 10 = "0 - 10"

10 - 20 = "11 - 20"

20 - 30 = "21 - 30"

30 - 50 = "31 - 50"

50 - 70 = "51 - 70"
```

```
70 - HIVALUE = ">70"
END

GRAPH AgeCategory GRAPHTYPE="Pie" TITLETEXT="Church Supper Attendees"
```

Example 3: A scatter graph is displayed showing age by time of supper.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
GRAPH TimeSupper Age GRAPHTYPE="Scatter XY"
```

Example 4: A graph showing the number of ill persons and vanilla eaters is displayed using the form for the Oswego outbreak investigation.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
FREQ Vanilla Ill STRATAVAR=Sex OUTTABLE=VanillaOut
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:VanillaOut
SELECT Vanilla = (+) OR Ill = (+)
SET Missing = (-)
DEFINE Vanilla2  YN
DEFINE Ill2 YN
IF Ill = (+) THEN
   ASSIGN Ill2 = Sex
END
IF Vanilla = (+) THEN
   ASSIGN Vanilla2 = Sex
END
GRAPH Ill2 Vanilla2 GRAPHTYPE="Bar" TITLETEXT="Number of Ill Persons and Vanilla Eate
rs by Sex" WEIGHTVAR=Count
```

Example 5: An Epi Curve showing the incubation time for an unknown pathogen is displayed. Note that the DIALOG=(-) parameter ensures the graph window does not appear. Instead, the graph is generated and sent to the Analysis output window. This option can be useful when running automated scripts since it does not require user interaction to continue processing commands.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
DEFINE Incubation NUMERIC
ASSIGN Incubation = MINUTES(TimeSupper, DateOnset) / 60
GRAPH Incubation GRAPHTYPE="Epi Curve" XTITLE="Incubation Period (Hours)"
YTITLE="Number of Cases"
```

IF THEN ELSE

Description

This command defines conditions and one or more consequences which occur if the conditions are met. An alternative consequence can be given after the ELSE statement. The ELSE will be executed if the first set of conditions is not true. If the condition is true, the first statement is executed. If the statement is false, the statement is bypassed. If an ELSE statement exists, it is executed instead. THEN is part of the If Condition statement. It starts the section of code executed when the If condition is true.

Syntax

```
IF <expression> THEN
   [command(s)]
END

IF <expression> THEN
   [command(s)]
ELSE
   [command(s)]
END
```

- The <expression> represents a condition that determines whether or not subsequent commands will be run. If the condition evaluates to true, the commands inside of the IF block will run. If the condition evaluates to false, the commands inside of the ELSE block will run instead. If no ELSE exists and the condition is false, then no commands inside of the IF block are run.
- The [command(s)] represents at least one valid command.
- The ELSE statement is optional and will run any code contained inside of it when the <expression> evaluates to false.

Comments

An IF statement is executed immediately if it does not refer to a database variable, if characteristic or attribute that can be measured, or if any defined variables have been assigned literal values. If the statement, YEAR = 97 has already occurred, then an IF statement dependent on it, such as IF YEAR = 97 then …., is executed immediately.

```
IF Age > 15 THEN
   ASSIGN Group = "ADULT"
ELSE
   ASSIGN Group = "CHILD"
END
```

It is important to cover all the conditions in IF statements to avoid gaps in the logic and results. Sometimes it is important to have an ELSE condition that covers conditions not included in other IF clauses. This effect is best achieved by setting the variable initially to something other than missing.

```
DEFINE ILL YN
ASSIGN ILL = (-)

IF Vomiting = (+) THEN

ASSIGN ILL = (+)
END

IF Diarrhea = (+) THEN
   ASSIGN ILL = (+)
END

IF Fever = (+) THEN
   ASSIGN ILL = (+)
END
Set Ill = (+) only if one or more symptoms are present.

The same result could be achieved with this code:

IF (Diarrhea = (+)) OR (Vomiting = (+)) OR (Fever = (+)) THEN
   ASSIGN ILL = (+)
ELSE
   ASSIGN ILL = (-)
END
```

Examples

Example 1: The Group variable for all records in the data set is assigned the value of "Young Adult" if the Age variable has a value between (but not including) 17 and 30. If the value in Age falls outside of this range, no value is assigned to Group.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
DEFINE Group TEXTINPUT
IF Age < 30 AND Age > 17 THEN
   ASSIGN Group = "Young Adult"
END

LIST Group
```

Example 2: Several different values are assigned to the Group variable depending on the value of the Age variable.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
DEFINE Group TEXTINPUT
IF Age < 30 AND Age > 17 THEN
   ASSIGN Group = "Young Adult"
END

IF Age <= 17 THEN
   ASSIGN Group = "Minor"
END

IF Age >= 30 THEN
   ASSIGN Group = "Adult"
END

LIST Group
```

Example 3: If the patient ate chocolate or vanilla ice cream, the variable IceCream is assigned a value of true. Otherwise, it is assigned a value of false.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
DEFINE IceCream YN

IF Vanilla = (+) OR Chocolate = (+) THEN
   ASSIGN IceCream = (+)
ELSE
   ASSIGN IceCream = (-)
END

LIST Vanilla Chocolate IceCream
```

KMSURVIVAL

Description

This command performs the Kaplan-Meier Survival Analysis and produces graphs and statistics for one or several groups of subjects being followed in a clinical study. At any given time, some of the subjects may be "censored," not having information available on their status. KMSurvival is especially constructed to deal with this situation. Statistics showing the risk set by group and time can be written to an OUTTABLE for later formatting.

Syntax

```
KMSURVIVAL <time variable>=<group variable> * <censor variable> (<value>)
[TIMEUNIT="<time unit>"] [OUTTABLE=<tablename>] [GRAPHTYPE="<graph type>"]
[WEIGHTVAR=<weight variable>]
```

- The <time variable> represents the variable specifying the time of the event.
- The <group variable> represents the variable that indicates to which treatment or control group the subject belongs.
- The <censor variable> represents the variable to describe an event as failure or censored.
- The <value> represents the value of the censor variable indicating uncensored (failure).
- The <time unit> represents a value for labeling the time axis.
- The <graph type> represents the following:
  - **Survival Probability** is the observed survival over the different groups.
  - **Log-Log Survival** is the log of the negative log of the survival probability.
  - **None** does not produce a graph. If no graph type is specified, the default Survival Probability curve is plotted.
- The <weight variable> represents a variable in the database that specifies the contribution or each data row to the output.

Example

The commands below show a comparison between two groups.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Addicts
KMSURVIVAL Survival_Time_Days = Clinic * Status ( 1 ) TIMEUNIT="Days"
```

LIST

Description

This command does a line listing of the current dataset. If variable names are given, LIST displays only these variables. LIST * displays all variables of all active records. LIST * EXCEPT displays all variables of all active records except those named.

Syntax

```
LIST {* EXCEPT} [<variable(s)>] LIST {* EXCEPT} [<variable(s)>] {GRIDTABLE}
```

- The * asterisk is used to represent all variables
- The <variable(s)> represents one or more variable names.
- The keyword GRIDTABLE specifies that data is displayed as a grid for viewing only, instead of HTML format.

Comments

Adding an EXCEPT Variable list excludes all the named variables from a LIST or LIST *.

If the dataset has been sorted with the SORT command, the records are listed in sorted order. Otherwise, they are listed in an order determined by the database.

Examples

Example 1: All variables are listed using the grid format.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
LIST * GRIDTABLE
```

Example 2: The Age variable is listed using the web (HTML) format.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
LIST Age
```

Example 3: The variables Sex, Vanilla, Chocolate, Ill, and Age are listed using the grid format.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
LIST Sex Vanilla Chocolate Ill Age GRIDTABLE
```

LOGISTIC

Description

This command performs conditional or unconditional multivariate logistic regression with automatic dummy variables and support for multiple interactions.

Syntax

```
LOGISTIC <dependent variable> = <independent variable(s)> [MATCHVAR=<match variable>]
[NOINTERCEPT] [OUTTABLE=<tablename>] [WEIGHTVAR=<weight variable>] [PVALUE=<PValue>]
```

- The <dependent variable> represents the dependent variable.
- The <independent variable(s)> represents an independent variable that can be a numeric variable, a non-numeric variable, or a variable surrounded by parenthesis. Any text or yes-no variable, or a variable surrounded with parenthesis, is automatically recoded into dummy variables. A dummy variable is created for all but one of the levels of a variable. The variable measures the contribution of its level relative to the excluded level. Interactions are specified by * between variables.
- The <weight variable> represents a variable to specify the contribution each data row has to the output.
- The <match variable> represents a variable to specify the different groups for a conditional analysis. If a match variable is specified, a conditional analysis will be performed
- The <tablename> represents a table where the residuals are stored. If no table name is present, no residuals are produced.
- The <PValue> represents the size of the confidence intervals; this may be specified as percent or decimal. If greater than .5, 1-PValue is used. The default is 95%.

Comments

LOGISTIC uses the Newton-Rhapson method for maximizing likelihood.

Example

To do an unconditional regression, run the following:

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
LOGISTIC Ill = BrownBread CabbageSal Water Milk Chocolate Vanilla
```

MEANS

Description

This command is used to compute descriptive statistics for a continuous (numeric) variable. When used with a cross-tabulation variable, it also computes statistics showing the likelihood that the means of the groups are equal. The mean of a yes-no variable is the proportion of respondents answering yes.

Syntax

```
MEANS <variable 1> {<variable 2>} {STRATAVAR=<variable(s)>} {WEIGHTVAR=<variable>}
{OUTTABLE=<tablename>}
```

- **<variable 1>** represents a numeric variable to be used to calculate means (or * for all numeric variables).
- **<variable 2>** represents any variable used for cross-tabulation (optional).
- **<variable(s)>** represent variable(s) to be used for stratified analysis.
- **<variable>** represents a variable containing the frequency for the event.
- **<tablename>** represents a name for a table to be created.

Comments

The MEANS command has two formats. If only one variable is supplied, the program produces a table similar to one produced by FREQUENCIES, plus descriptive statistics. If two variables are supplied, the first is a numeric variable containing data to be analyzed and the second is a variable that indicates how groups will be distinguished. The output of this format is a table similar to one produced by TABLES, plus descriptive statistics of the numeric variable for each value of the group variable.

Multiline (memo) variables cannot be used in MEANS. To use a Multiline variable, define a new variable and assign to it the value SUBSTRING(<old variable>,1,255) and use it in the means.

The f-test which is generated from MEANS is a generalization of the t-test. The t-test only works with two groups while the f-test works with any number of groups.

MEANS produces the following statistical tests:

- Parametric
- ANOVA (for two or more samples)
- Student's t-test (for two samples)
- Non-parametric
- Kruskal-Wallis one-way analysis of variance (for two or more samples)

- Mann-Whitney U = Wilcoxon Rank Sum Test (for two samples)

Examples

Example 1: Descriptive statistics for the age variable are displayed, including the number of observations, the total, the mean, variance, standard deviation, 25%, median, 75%, maximum, minimum, and mode.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
MEANS Age
```

Example 2: The MEANS command is used to compare two means. An independent t-test and one-way analysis of variance (ANOVA) is performed.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:EvansCounty
MEANS CHL CHD
```

Example 3: The average number of cigarettes smoked between males and females is determined.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Smoke
MEANS NumCigar Sex STRATAVAR=Strata WEIGHTVAR=SampW PSUVAR=PSUID
```

MERGE

Description

This command merges records in one dataset with those in another the Global Record Identifier contained in every Epi Info 7 project to establish the match between records. Records in the second dataset that do not have matching keys can be appended to the end of a dataset. Records in the second dataset that do have matching keys can be used to update records in the main dataset. Records in the second dataset can be used as the parent in defining an Epi Info 7 parent-child relationship after records have been merged into the parent and child forms.

Syntax

```
MERGE <table specification> {LINKNAME=<text>} [<key(s)>] <type>
```

- The <table specification> represents the type and name of a data table to be read as the Merge file.

  The <key(s)> represent the Global Record Identifier on which the match or relate will be performed. These are in the form: <ExpressionCurrntTable>::<ExpressionMergeTable>
- The <MergeType>, may be APPEND, UPDATE or RELATE. If no type is specified, APPEND and UPDATE are performed. For matching records, UPDATE replaces the value of any field in the READ table whose name matches in the MERGE table with the value from the MERGE table unless it has a missing value. For unmatched records, APPEND creates a new record in the READ table with values only for those fields which exist in the MERGE table.
- Currently, Merge is only supported when the READ and MERGE data source is an Epi Info 7 project.

Comments

APPEND adds unmatched records in the Merge table to the currently active dataset. Only fields found in both datasets are added.

UPDATE replaces fields of records in the active table with those in the Merge table if the key expressions match. Only fields found in both datasets with a non-empty value in the Merge table are replaced.

RELATE moves the unique key of the current table to the foreign key of the related table to make a permanent relationship. The related (Merge) table must be an Access/Epi2000 table. The READ table and the RELATE tables must be Epi Info forms. READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:

Example

```
MERGE {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Surveillance GlobalRecordId ::
GlobalRecordId
```

PRINTOUT

Description

This command sends the contents of the current output file (the one visible in the output window) or some other specified file to the default printer.

Syntax

```
PRINTOUT <filename>
```

Example

The output file Oswego.txt is sent to the printer.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
WRITE REPLACE "Text" 'C:\Epi_Info\Oswego.txt' *
PRINTOUT 'C:\Epi_Info\Oswego.txt'
```

QUIT

Description

This command closes the current data files and terminates the current program, closing Classic Analysis.

Syntax

```
QUIT
```

Program Specific Features

Quit will stop the execution of a program and close Classic Analysis. If there is no QUIT at the end of a .PGM program, Classic Analysis continues to run and offer user-interaction.

Example

You are presented with a dialog box asking if you want to close Analysis. Selecting 'Yes' closes Classic Analysis.

```
DEFINE Results YN
DIALOG "Do you wish to close Analysis?" Results YN
IF Results = (+) THEN
   QUIT
END
```

READ

Description

This command makes one or more forms the active dataset. It also removes any previously active datasets and associated defined variables and dataset-specific commands (e.g., RELATE, SORT, SELECT or IF statements).

Syntax

```
READ <table specification> FMT="<file format>" UID="<username>" PWD="<password>" END
```

The <table specification> and FILESPEC are referred to in the following paragraphs:

The READ, RELATE, and MERGE commands can operate on many different types of data. Each type requires a different table specification, and some types required additional information in a file specification. Table specifications usually consist of a data type (contained in double quotes), a space, a file path (which may be enclosed in single quotes, and must be if it includes a space), a colon, and a table name.

The various file types that can be used are:

- Epi Info 7 Forms and Tables
- Microsoft Access 97-2003 and MS Access 2007
- Microsoft Excel 97-2003 and Excel 2007
- **SQL Server** – Server name and Database name will be required when accessing tables within an SQL Server database.
- **Text Files** – There are two basically different forms of text files. Both forms have only one table per file, so there is no need to specify a table. Both forms put the data for one record on a single line. The difference is in how the fields are indicated. One form, called "delimited," uses designated characters to separate fields. The second form, shown in the fourth example, is called "fixed" because each field occupies the same positions in each line. All character positions through the last field must be accounted for even if they do not contain useful data (the command generator will automatically generate filler fields as required). Even if the first line of the file contains field names (HDR="YES"), the names specified in field definitions will be used. The text file driver actually reads the file into the database, so changes made to the file after the READ will not be saved and changes made through Epi Info will not be saved to the text file (unless it is rewritten with the WRITE REPLACE command, which is available only in CSV format).

Examples

Example 1: If the file/form was previously accessed, the Oswego Form is read

```
READ {config:Sample.prj}:Oswego
```

Example 2: The Oswego Form is read. Unlike in example 1, the full path to the database file is specified.

```
READ {C:\Epi Info 7\Epi Info 7\Projects\Sample\Sample.prj}:Oswego
```

Example 3: If a space appears in the table name in Access, it must be enclosed in square brackets.

```
READ {C:\Epi Info 7\Epi Info 7\MyData}:[Table Name with Spaces]
```

Example 4: An Excel spreadsheet is read.

```
READ {C:\Epi Info 7\Epi Info 7\PlagueData.xls}:[Plague$]
```

RECODE

Description

This command is used to change some or all the values of a variable. New values can be stored in the same variable or in a new one. It can also be used to convert a numeric variable into a character variable or the reverse, or to create a new variable based on recoded values of an existing variable.

Syntax

```
RECODE <variable1> TO <variable2>

   value1 - value2 = <recoded value>

   value1 – HIVALUE = <recoded value>

   LOVALUE - value2 = <recoded value>

   value3 = <recoded value>

ELSE = <recoded value>

END
```

- The <variable1> represents the donor variable (where the values are).
- The <variable2> represents the receiver variable (where recoded values will be).

Comments

Text values must be enclosed in quotation marks; numeric, date; yes/no values must not. All recoded values must be of the same type. Numeric ranges are separated by a space, hyphen, and space, as in 1 - 5. Negative values are permitted (i.e.,-10, -9, and -8). The words LOVALUE and HIVALUE may be used to indicate the smallest and largest values representable in the database. The word ELSE may be used to indicate all values not falling in the preceding ranges. Recodes take place in the order stated; if two ranges overlap, the first in order will apply. Analysis cannot RECODE more than about 12 levels of values. If this is a problem, do as many recodes as possible, write a new table, READ it, and do more recodes.

Examples

Example 1: The RECODE command is used to generate an age range.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
DEFINE AgeRange TEXTINPUT
RECODE Age TO AgeRange LOVALUE - 0 = "<=0" 0 - 10 = ">0 - 10"= 10 - 20 = ">10 - 20" 20
- 30 = ">20 - 30" 30 - 40 = ">30 - 40"

40 - 50 = ">40 - 50"

50 - 60 = ">50 - 60"
```

```
60 - 70 = ">60 - 70"

70 - 80 = ">70 - 80"

80 - 90 = ">80 - 90"

90 - 99 = ">90 - 99"

99 - HIVALUE = ">99"
END
LIST Age AgeRange
```

Example 2: The RECODE command is used to generate an age range. The ELSE clause ensures that any values not captured in the recoding process are assigned a default value. In this case, any values greater than 60 are assigned "Senior."

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
DEFINE AgeRange TEXTINPUT
RECODE Age TO AgeRange

LOVALUE - 0 = "<=0"

0 - 10 = ">0 - 10"

10 - 20 = ">10 - 20"

20 - 30 = ">20 - 30"

30 - 50 = ">30 - 50"

50 - 65 = ">50 - 65"
ELSE = "Senior"
END
LIST Age AgeRange
```

Example 3: The RECODE command is used to generate a detailed age range from 0 to 70 in increments of three. Note that a single RECODE command is limited to approximately 12 conditions because of query size limitations inherent in the Access database format. The desired age categories would require more than 12 recodes. To work around this problem, only 10 recodes are done at a time and are separated by a series of SELECT, WRITE, and READ commands. The first WRITE command creates a new temporary table (or overwrites an existing one) that stores only records that contain recoded values. The remaining records are not written out because of the SELECT command. Each subsequent block of recoded values is written to the same file using the APPEND parameter. By the time the code is done executing, the table T1 contains all of the recoded data.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
DEFINE AgeRange TEXTINPUT
RECODE AGE TO AgeRange
LOVALUE - 0 = "<=0"
0 - 3 = ">0 - 3"
3 - 6 = ">3 - 6"
6 - 9 = ">6 - 9"9 - 12 = ">9 - 12"
12 - 15 = ">12 - 15"
```

```
15 - 18 = ">15 - 18"
18 - 21 = ">18 - 21"
21 - 24 = ">21 - 24"
24 - 27 = ">24 - 27"
END

SELECT NOT AgeRange = (.)
WRITE APPEND "Epi7" {Provider=Microsoft.Jet.OLEDB.4.0;Data
Source="{C:\Epi_Info_7\Projects\Sample\Sample.prj}: T1 *

READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
DEFINE AgeRange TEXTINPUT
RECODE Age TO AgeRange
27 - 30 = ">27 - 30"
30 - 33 = ">30 - 33"
33 - 36 = ">33 - 36"
36 - 39 = ">36 - 39"
39 - 42 = ">39 - 42"
42 - 45 = ">42 - 45"
45 - 48 = ">45 - 48"
48 - 51 = ">48 - 51"
51 - 54 = ">51 - 54"
54 - 57 = ">54 - 57"
END

SELECT NOT AgeRange = (.)
WRITE APPEND "Epi7" {Provider=Microsoft.Jet.OLEDB.4.0;Data
Source="{C:\Epi_Info_7\Projects\Sample\Sample.prj}: T1 *

READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
DEFINE AgeRange TEXTINPUt
RECODE Age TO AgeRange
57 - 60 = ">57 - 60"
60 - 63 = ">60 - 63"
63 - 66 = ">63 - 66"
66 - 69 = ">66 - 69"
69 - 70 = ">69 - 70"
70 - HIVALUE = ">70"
END

SELECT NOT AgeRange = (.)
WRITE APPEND "Epi7" {Provider=Microsoft.Jet.OLEDB.4.0;Data
Source="{C:\Epi_Info_7\Projects\Sample\Sample.prj}: T1 *
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:T1
LIST Age AgeRange
```

REGRESS

Description

This command performs multivariate linear regression with support for automatic dummy variables and multiple interactions. If a variable has more than two values, it is automatically turned into a series of 'yes/no' variables called 'dummy variables', one for each extra value.

Syntax

```
REGRESS <dependent variable> = <independent variable(s)> [NOINTERCEPT]
[OUTTABLE=<tablename>] [WEIGHTVAR=<weight variable>] [PVALUE=<PValue>]
```

- **<dependent variable>** represents the dependent variable.
- **<independent variable(s)>** is an independent variable that can be a numeric variable, a non-numeric variable, or a variable surrounded by parenthesis. Any text or yes-no variable, or a variable surrounded with parenthesis, is automatically recoded into dummy variables. A dummy variable is created for all but one of the levels of a variable. This dummy variable measures the contribution of its level relative to the excluded level. Interactions are specified by * between variables.
- **<weight variable>** represents a variable describing each data row's contribution to the regression
- **<tablename>** is the table to store the residuals. If no table name is present, no residuals are produced.
- **<PValue>** represents the size of the confidence intervals; may be specified as percent or decimal; if greater than .5, 1-PValue is used. The default is 95%.

Comments

REGRESS uses the least-squares method for determining coefficients.

Example

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:BabyBloodPressure
REGRESS SystolicBlood = AgeInDays Birthweight
```

RELATE

Description

This command links one or more tables to the current dataset during analysis, using a common identifier to find matching records. The identifier may span several fields, in which case values in each of the fields must match.

Syntax

```
RELATE <table specification> [<key(s)>] {ALL}
```

- The <table specification> represents the type and name of a data table to be read as the Related file.
- The <key(s)> represent one or more expressions that designate keys on which the match or relate will be performed. When multiple expressions are used, they are connected with AND. These following are in the form:

<ExpressionCurrentTable> :: <ExpressionRelateTable>

- The ALL represents records in the table(s) already READ or related for which there is no corresponding record in the RELATE table will be included in the data with null values for variables in the RELATE table. If absent, only records with corresponding records in the RELATE table will be included in the data.

Program Specific Features

Analysis

If the relationship was created in Form Designer, Classic Analysis can relate the two tables without need of a key expression.

Form Designer

- Grid tables and relate buttons help create parent-child relationships.
- Grid tables have a prefix of recgrid table.

Comments

To use RELATE, at least one table must have been made active with the READ command. The table to be linked must have a key field that identifies related records in the other table. In Epi Info 7, the keys in the main and related tables or files might not have to have the same name.

The expressions in the key are the names of variables in the tables that will be used to determine which records match each other. More than one key pair ("multiple keys") can be designated, separated by AND.

After issuing the RELATE command, the variables in the related table may be used as if they were part of the main table. Where variable names are duplicated in the related tables, the variable names will be suffixed with a sequence number. In referring to a variable in a related table, you may (optionally) use the form HOUSE.AGE to represent the variable AGE in the form HOUSE. This will distinguish it from another variable AGE that might be in the main table.

Frequencies, cross-tabulations, and other operations involving data in both the main and related tables can be performed. To preserve the linked structure, the WRITE command may be used to create a new table. More than one table may be related to the main table or related table by using successive RELATE commands.

Example

The records from the RHepatitis data tables are related to the Surveillance data table.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Surveillance

RELATE RHepatitis GlobalRecordId :: [FKEY]
```

ROUTEOUT

Description

This command directs output to the named file until the process is terminated by CLOSEOUT. Output from commands (e.g., FREQ and LIST) is appended to the same output file as it is produced.

Syntax

```
ROUTEOUT <filename> {REPLACE | APPEND}
```

- The <filename> represents an HTM document where the output will be stored. If no directory is specified, use the directory of the current project.
- The REPLACE | APPEND keyword controls what happens to an existing file with the same name. If REPLACE is specified, any existing file of the same name is deleted prior to writing. If APPEND is specified, new output is appended to any existing file with the same name.

Comments

Epi Info 7 sends output to an HTML document that any browser can be read. If no output is selected, Epi Info 7 creates a new file (typically called OUTXXX.HTM) where XXX is a sequential number. Output files are placed in the same directory as the current project. The prefix for output files and a starting sequence number can be changed from the Storing Output command located in the Output folder.

If no path is specified, or if the directory does not exist, the output file is created in the directory of the current project.

Example

The output generated by running the commands below is sent to the file Outbreak1.htm in the C:\Epi_Info_7\Projects\folder.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
ROUTEOUT "C:\Epi_Info_7\Projects\Outbreak1.htm" REPLACE
SET STATISTICS=COMPLETE
TABLES Vanilla Ill STRATAVAR=Sex
MEANS Chocolate Ill
CLOSEOUT
```

RUNPGM

Description

This command runs a stored Classic Analysis program. This command is similar to an INCLUDE file or subroutine in other systems.

Syntax

```
RUNPGM '<file name>':"<program>"
```

```
RUNPGM '<file name>'
```

- The <file name> represents the path and filename for the MDB or PGM file where the program is stored. If the path or filename contains a space, it must be enclosed in single quotes. If the program to be run is in the current project, the path need not be supplied.
- The <program> represents the Program name. If the program name contains a space, it must be enclosed in double quotes. This is not used for text files.

Comments

Since the filename can include any path and database name, the program to be executed can be stored in a different database.

Examples

Example 1: The program editor code contained in the Statistics.pgm file is run.

```
RUNPGM "C:\MyProject_Folder\Sample\Sample.prj":Statistics
```

SELECT

Description

This command allows an expression to be specified that must be true to process a record. It can also be called a data filter. If the current selection is Age>35, then only those records with age greater than 35 are selected. SELECT used alone without an expression cancels all previous SELECT statements. SELECT statements are cumulative until canceled. Therefore, Select Age > 34, Select Sex = “Male" will choose males over the age of 34. Cancel Select before doing Select Sex = “Female" because you will get only records that are male and female, or none at all.

Syntax

```
SELECT <expression>
```

- The <expression> represents any valid Epi Info 7 expression.

Comments

SELECT expressions are cumulative so that the two expressions:

SELECT Age > 35

SELECT Sex = "F"

are equivalent to

SELECT (Age > 35) AND (Sex = "F")

Examples

Example 1: A subset of the data contained in the Food History table is selected. In this case, only records where the patient is female and interviewed after 04/15/2011, or where the patient is male and interviewed after 06/01/2011 are selected to be in the subset. Note the use of parentheses to show relationships.

```
READ {C:\Epi_Info_7\Projects\Ecoli\EColi.prj}:FoodHistory
SELECT ((DateofInterview > 04/15/2011) AND (Sex =”F-Female")) OR ((DateofInterview >
05/15/2011) AND (Sex = "M-Male"))
LIST * GRIDTABLE
```

Example 2: A subset of the data contained in the Food History data table is selected. In this case, only records where the patient's first name is "Pam" are selected to be in the subset. Note that "Huber" is not case sensitive, and the SELECT command would have also selected "HUBER" and "huber”.

```
READ {C:\Epi_Info_7\Projects\Ecoli\EColi.prj}:FoodHistory
SELECT LastName = "Huber"
LIST * GRIDTABLE
```

Example 3: A subset of the data contained in the Food History data table is selected. In this case, only records where the patient's last name starts with the letters "Sch" and ends with the letter "r" are selected to be in the subset.

```
READ 'C:\Epi_Info\Refugee.mdb':Patient
SELECT LastName LIKE "Sch*r"
LIST LastName
```

Example 4: A subset of the data contained in the Oswego data table is selected. In this case, only records where the patient is ill are selected to be in the subset.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
SELECT Ill = (+)
LIST Name Age Ill
```

Example 5: A subset of the data contained in the Oswego data table is selected. In this case, all records except those that do not have a value for the TimeSupper variable (the field was left blank during data entry) will be selected to be in the subset.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
SELECT NOT TimeSupper = (.)
LIST TimeSupper DateOnset Sex Ill
```

Example 6: A subset of the data contained in the Oswego data table is selected. Two SELECT commands have a cumulative effect so that the two expressions are equivalent to SELECT (Age > 30) AND (Sex = "Male"). Only records where the age is greater than 30 and the sex is male will be selected.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
SELECT Age > 30
SELECT Sex = "Male"
LIST Age Sex
```

Example 7: A subset of the data contained in the Oswego data table is selected. Two SELECT commands have a cumulative effect making the two expressions equivalent to SELECT (Age > 30) AND (Age < 20). Only records where the age is greater than 30 and less than 20 are selected. Because these two conditions are mutually exclusive, the LIST command produces no output. A CANCEL SELECT command may be issued to clear the current selection criteria.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
SELECT Age > 30
SELECT Age < 20
LIST Age Sex
```

SET

Description

This command provides various options that affect the performance and output of data in Classic Analysis. These settings are utilized whenever you use the Classic Analysis program.

Syntax

```
SET [<parameter> = <value>]
```

The <parameter> = <value> represents various elements on the form. Any number of elements may be used in a single SET statement.

Parameter	Values	Response
(-)	"<Text>"	In Boolean variables, NO will be represented as <Text>.
(.)	"<Text>"	Variables with missing values will be represented as <Text>.
(+)	"<Text>"	In Boolean variables, YES will be represented as <Text>.
YN	"<Text1>"	Sets displayed text for Yes, No, and Missing to Text1, Text2, and Text3 respectively.
PROCESS	NORMAL	Deleted records are not included.
	DELETED	Only deleted records are included.
	BOTH	Deleted records are included.
DELETED	YES or (+)	Deleted records are included.

Parameter	Values	Response
	NO or (-)	Deleted records are not included.
	ONLY	Only deleted records are included.
MISSING	(+) or ON	Include missing values for analysis.
	(-) or OFF	Do not include missing values for analysis.

Program Specific Feature

In the command generator, selecting Save All generates a SET command, which contains all current settings. Selecting Save Only or OK generates a SET command, which contains only changed settings. To force a SET command to be generated for a current value of a setting, change its value and change it back again.

Example

```
SET MISSING=(-)
```

SORT

Description

This command allows a sequence to be specified for records to appear in LIST, GRAPH, and WRITE commands. If no variable names are specified after the SORT command, the current sort is cleared, and subsequent record outputs will be in the order of the original data table. If one variable name is given, records are sorted using that variable as the "key". If more than one variable is specified, records will be put in order by the first variable, then within a group with the same value of variable 1, the ordering will be by variable 2, etc.

Syntax

```
SORT <variable> {DESCENDING}
```

- <variable> represents the variable to be sorted by.
- DESCENDING indicates that the sort order is descending; if not specified, ascending order will be used.

Comments

The parameter DESCENDING must be placed next to the variable to be sorted in descending order. Should several variables be sorted in descending order, one DESCENDING should be included for each of them.

Examples

Example 1: The data is sorted by Age in ascending order.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
SORT Age
LIST Age Sex Ill GRIDTABLE
```

Example 2: The data is sorted by Age in descending order. If two or more records have the same value for Age, the records are sorted by Ill in descending order. If two or more records have the same value for Ill, the records are sorted by Sex in descending order.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
SORT Age DESCENDING Ill DESCENDING Sex DESCENDING
LIST Age Sex Ill GRIDTABLE
```

SUMMARIZE

Description

This command creates a new table containing summary statistics for the current dataset or its strata.

Syntax

```
SUMMARIZE varname::aggregate(variable) [varname::aggregate(variable) ...] TO tablename
STRATAVAR=variable list {WEIGHTVAR=variable}
```

- Available aggregates are COUNT, MIN, MAX, SUM, FIRST, LAST, AVG, VARiance and STandardDEViation (Sum, Avg, Var and StDev available only for numeric fields). COUNT may be used without a variable in parenthesis to indicate that a count of the number of records in the table or strata is desired. You can also use COUNT with a variable in parenthesis to indicate that the number of records in the table or strata with non-missing values of the specified group is desired. FIRST and LAST are based on the current sort order.

Comments

Classic Analysis creates a new table or appends to an existing table (tablename) containing variables (varname) which represent aggregates of variables in the current data source (aggregate[variable]). The aggregates are computed for each group of records, determined by the STRATAVARs, which are also included in the table. Available aggregates are COUNT, MIN, MAX, SUM, AVG, VARiance and STandardDEViation (Sum, Avg, Var and StDev available only for numeric fields). COUNT may be used without a variable in parenthesis to indicate that a count of the number of records in the group is desired, or with a variable in parenthesis to indicate that the number of records in the group with non-missing values of the specified group is desired.

This command solves some recurring problems for programmers. One is computing percents; it is difficult to get a denominator. Another is determining the earliest or latest date in a list of relevant dates, or the highest or lowest of a series of measurements. Many problems can be solved with the OUTTABLE from a TABLES or FREQ command, or with self-joins, but this provides a straightforward method to achieve these results.

Note: Multi-line (memo) fields are not permitted.

Example

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:EvansCounty
SUMMARIZE Average_Age :: Avg(AGE) Average_DBP :: Avg(DBP) Number_Records :: Count(AGE)
Std_Age :: StDev(AGE) Std_DBP :: StDev(DBP) TO SUMMARY_TABLE
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Summary_Table
LIST *
```

TABLES

Description

This command cross-tabulates specified variables in two or three dimensions. Values of the first variable appear across the top of the table while those of the second variable appear in the left margin of the table. Unique values of additional variables are represented as strata. Normally, cells contain counts of records matching the values in the corresponding marginal labels. If a WEIGHTVAR parameter is given, the cells represent sums of the weight variable. TABLES DISEASE COUNTY WEIGHTVAR=COUNT provide the same results as SUMTABLES COUNT DISEASE COUNTY in Epi6.

Syntax

```
TABLES <exposure> <outcome> {STRATAVAR=[<variable(s)>]} {WEIGHTVAR=<variable>}
{PSUVAR=<variable>} {OUTTABLE=<table>}
```

- **<exposure>** represents the variable in the database to be considered the risk factor (or * for all variables).
- **<outcome>** represents the variable in the database considered disease of consequence (or * for all variables).
- **<variable>** represents the variable in the database.
- **<table>** represents a valid table name to be used to store output.

Comments

For every possible combination of values of the strata variables, a separate table (stratum) for variable 1 by variable 2 is produced. TABLES BAKEDHAM ILL STRATAVAR=SEX produces a table of BAKEDHAM by ILL for each value of sex-one for M and one for F. TABLES BAKEDHAM ILL STRATAVAR=SEX RACE produces a separate table of BAKEDHAM by ILL for each combination of SEX and RACE-female/black, female/white, male/black, male/white, etc.

If * is given instead of a variable name, each variable in the dataset is substituted for * in turn. To analyze each variable by illness status, use the command TABLES * ILL which produces tables of SEX by ILL, AGE by ILL, etc.

It is important to consider using * or requesting multidimensional tables if the dataset is large (thousands of records), since it may produce more tables than needed in terms of time, paper, and other costs. Press **Ctrl-Break** to exit from a lengthy procedure.

For 2x2 tables, the command produces odds and risk ratios. For these values to have their accepted epidemiological meanings, the value representing presence of the exposure and outcome conditions must appear in the first row and column of the table. Epi Info yes/no variables are automatically sorted properly. The STATISTICS setting controls the detail to which the statistics are reported. For tables other than 2x2, Chi-square statistics are computed. If an expected value is < 5, the message Chi-square not valid appears. The mid-p and Fisher exact are preferred, especially in this case.

Multiline (memo) variables cannot be used in tables. To use a Multiline variable, define a new variable and assign to it the value SUBSTRING(<old variable>,1,255) and use it in the table.

Examples

Example 1: A 2x2 table is generated showing coronary heart disease (CHD) by Catecholamine Level (CAT).

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:EvansCounty
TABLES CAT CHD
```

Example 2: A 2x2 table is generated showing coronary heart disease (CHD) by Catecholamine Level (CAT), stratified by an age group variable of type yes/no.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:EvansCounty
TABLES CAT CHD STRATAVAR=AgeG1
```

Example 3: A 2x2 table is generated for every variable in the database using Ill as the outcome variable for table.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
TABLES * Ill
```

Example 4: A 2x2 table is generated and saved to a separate table in the Sample database using the OUTTABLE parameter.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
TABLES Vanilla Ill OUTTABLE = T1
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:T1
LIST * GRIDTABLE
```

Example 5: A 2x2 table is generated showing obesity and disease outcome. The analysis is weighted by the value contained in the COUNT column.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Lasum
TABLES OB OUTCOME WEIGHTVAR=COUNT
```

Example 6: A complex sample table is generated using a stratified cluster survey.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Epi10
TABLES Prenatal VAC STRATAVAR=Location WEIGHTVAR=POPW PSUVAR=Cluster
```

TYPEOUT

Description

This command inserts text, a string or file contents, into the output. Typical uses might include comments or boilerplate.

Syntax

```
TYPEOUT "text" ([<font property>]) {TEXTFONT <color> <size>}
```

- **<text>** represents the text to be displayed.
- **<font property>** represents the properties Underline, Bold, or Italic separated by commas.
- **<color>** represents Aqua, Lime, Red, Black, Silver, Maroon, Blue, Navy, Teal, Fuchsia, Olive, White, Grey, Purple, Yellow, or Green. Color can also be represented by hexadecimal digits ###### with pairs of digits representing the amount of red, green, and blue on a scale of 0–255.

Comments

TYPEOUT with a text string is similar to TITLE except that TYPEOUT places the text once when the command is encountered. HEADER appears at the top of each segment of output until cleared.

If no text in quotation marks follows the TYPEOUT command, TYPEOUT sends the file contents specified to the current output. It can be in text or HTML.

Examples

Example 2: Displays the word "Confidential" underlined and in bold.

```
TYPEOUT "Confidential" (BOLD,UNDERLINE)
```

UNDEFINE

Description

This command removes a defined variable and any assigned values from the system.

Syntax

```
UNDEFINE <variable>
```

- The <variable> represents a defined variable.

Program Specific Feature

Permanent variables cannot be undefined from the Undefine dialog box. To undefine a permanent variable, type the syntax into the Program Editor and run the command.

Comments

A variable that already exists in the database cannot be undefined. To remove a database variable from the database, use the WRITE command with the EXCEPT modifier.

Example

```
UNDEFINE NewVar
```

UNDELETE

Description

This command will mark logically deleted records as normal.

Syntax

```
UNDELETE *

UNDELETE expression
```

Comments

Undelete * or expression causes all logically deleted records in the current selection matching the expression to be set to normal status. This applies only to Epi Info 7 forms and may not be used when using related tables.

Example

```
UNDELETE AgeInDays > 5
```

WRITE

Description

The WRITE command sends records to an output table or file in the specified format. Specifications include which variables are written, variable order, and the type of file to be written.

Syntax

```
WRITE <METHOD> {<output type>} {<project>:}table {[<variable(s)>]}

WRITE <METHOD> {<output type>} {<project>:}table * EXCEPT {[<variable(s)>]}
```

- **<METHOD>** represents either REPLACE or APPEND
- **<project>** represents the path and filename of the output.
- **<variable(s)>** represents one or more variable names.
- **<output type>** represents the following allowable outputs:

Database Type	Specifier	Element
Epi Info 7	"Epi Info 7”	<path:<table>
MS Access 97-2003	MS Access 97-2003	<path>
MS Access 2007	MS Access 2007	<path>
Excel 97-2003	MS Access 97-2003	<path>
Excel 2007	MS Access 2007	<path>
SQL Server		Server Name & Database Name
Text (Delimited)	"Text"	<path>

Comments

Records deleted in Enter or selected in Classic Analysis are handled as in other Analysis commands. Defined variables may be written to allow you to create a new Epi Info 7 file to make permanent changes. Unless explicitly specified, global and permanent variables will not be written.

To write only selected variables, the word EXCEPT may be inserted to indicate all variables except those following EXCEPT.

If the output file specified does not exist, the WRITE command will attempt to create it.

Either APPEND or REPLACE must be specified to indicate that an existing file/table by the same name will be erased or records will be appended to the existing file/table. If not all of the fields being written match those in an existing file during an APPEND, the unmatched fields are added to the output table.

Examples

Example 1: The Oswego data table (75 records) from Sample.PRJ is written to a database called SampleOutput. The destination table name is called Oswego_1. The second READ command reads the newly-created data table.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
WRITE REPLACE "Epi 2000" 'C:\Epi_Info\SampleOutput.mdb':Oswego_1 *
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego_1
```

Example 2: The Oswego data table (75 records) from Sample.PRJ is written to a database called SampleOutput three times. The APPEND method ensures that each WRITE command appends the entire data set several times. After all three WRITE commands have been run, the Oswego_2 table inside SampleOutput.mdb will contain 225 records.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
WRITE APPEND "Epi7" {Provider=Microsoft.Jet.OLEDB.4.0;Data
Source="C:\Epi_Info_7\Projects\SampleOutput.mdb"} : OSWEGO_2 *
WRITE APPEND "Epi7" {Provider=Microsoft.Jet.OLEDB.4.0;Data
Source="C:\Epi_Info_7\Projects\SampleOutput.mdb"} : OSWEGO_2 *
WRITE APPEND "Epi7" {Provider=Microsoft.Jet.OLEDB.4.0;Data
Source="C:\Epi_Info_7\Projects\SampleOutput.mdb"} : OSWEGO_2 *

READ {C:\Epi_Info_7\Projects\SampleOutput.mdb”}:OSWEGO_2
```

Example 3: This example shows how to make defined variables into permanent database variables. The Oswego data table from Sample.PRJ is written to a database called SampleOutput. Notice that the defined variable IncubationTime does not exist in Sample.PRJ, but after the WRITE command has executed, it is now a part of the newly-created data table Oswego_3.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
DEFINE IncubationTime NUMERIC
ASSIGN IncubationTime = HOURS(TimeSupper, DateOnset)
LIST IncubationTime
WRITE REPLACE "Epi7" {Provider=Microsoft.Jet.OLEDB.4.0;Data
Source="C:\Epi_Info_7\Projects\SampleOutput.mdb"} : OSWEGO_3 *
READ {C:\Epi_Info_7\Projects\SampleOutput.mdb”}:OSWEGO_3


LIST * GRIDTABLE
```

Example 4: The records from the Oswego Form in the Sample database are exported to an Excel spreadsheet. By specifying age, sex, and incubation time in the WRITE command, only those variables will be exported. This example may not work if Microsoft Excel is not installed on the computer.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
DEFINE IncubationTime
ASSIGN IncubationTime = HOURS(TimeSupper, DateOnset)
WRITE REPLACE "Epi7" {Provider=Microsoft.Jet.OLEDB.4.0;Data
Source="C:\Epi_Info_7\Projects\Oswego.xls";Extended Properties="Excel
8.0;HDR=Yes;IMEX=1"} : OSWEGO_1 Age Sex IncubationTime*
READ {C:\Epi_Info_7\Projects\Oswego.xls}:[OSWEGO_1$]

LIST * GRIDTABLE
```

Example 5: The records from the Oswego Form in the Sample database are exported to a text file.

```
READ {C:\Epi_Info_7\Projects\Sample\Sample.prj}:Oswego
WRITE REPLACE "Epi7" {Provider=Microsoft.Jet.OLEDB.4.0;Data Source="C:\
My_Project_Folder ";Extended Properties="text;HDR=Yes;FMT=Delimited"} : [OSWEGO#txt] *
```

THIS PAGE IS

INTENTIONALLY BLANK.

14. Functions and Operators

Introduction

Functions modify the value of one or more variables to produce a result (i.e., ROUND(2.33333) produces the value 2). Operators are used to combine two items (i.e., the + operator combines Var1 and Var2 to produce a sum, as in Var3=Var1+Var2). Functions and operators appear within commands and are used for common tasks that include extracting a year from a date, combining two numeric values, or testing logical conditions. Almost all functions require arguments enclosed in parentheses and separated by commas. If arguments are required, do not place any spaces between the function name and the left parenthesis. Syntax rules must be followed. Quotes that must enclose text strings are displayed in question or prompt dialog boxes. Parentheses must enclose arithmetic expressions and can explicitly control the order of operations. Parentheses also enclose function arguments.

Syntax Notations

The following rules apply when reading this manual and using syntax:

Syntax	Explanation
ALL CAPITALS	Epi Info commands and reserved words are shown in all capital letters similar to the READ command.
<parameter>	A parameter is information to be supplied to the command. Parameters are enclosed with less-than and greater-than symbols or angle brackets < >. Each valid parameter is described following the statement of syntax for the command. Parameters are required by the command unless enclosed in braces { }. Do not include the < > symbols in the code.
[<variable 1>]	Brackets [ ] around a parameter indicates that there can potentially be more than one parameter.
{<parameter>}	Braces { } around a parameter indicate that the parameter is optional. Do not include the { } symbols in the code.

Syntax	Explanation
|	The pipe symbol '|' is used to denote a choice and is usually used with optional parameters. An example is in the LIST command. You can use the GRIDTABLE or the UPDATE option, but not both. The syntax appears as follows with the pipe symbol between the two options: LIST {* EXCEPT} <VarNames> {GRIDTABLE | UPDATE}
/* */	The combination of backslash and asterisk in the beginning of a line of code and an asterisk and backslash, as shown in some code samples, indicates a comment. Comments are skipped when a program is run.
" "	Quotation marks must surround all text values as in: `DIALOG "Notice: Date of birth is invalid."`

Operators

There are various types of operators discussed in this appendix. The following types are provided:

- **Arithmetic Operators** are used to perform mathematical calculations.
- **Assignment Operators** are used to assign a value to a property or variable. Assignment Operators can be numeric, date, system, time, or text.
- **Comparison Operators** are used to perform comparisons.
- **Concatenation Operators** are used to combine strings.
- **Logical Operators** are used to perform logical operations and include AND, OR, or NOT.
- **Boolean Operators** include AND, OR, XOR, or NOT and can have one of two values, true or false.

Operator Precedence

If several operations occur in an expression, each part is evaluated and resolved in a predetermined order called Operator Precedence. Parentheses can be used to override the order of precedence and evaluate some parts of an expression before others. Operations within parentheses are always performed before those outside. Within parentheses, however, normal Operator Precedence is maintained.

If expressions contain operators from more than one category, arithmetic operators are evaluated first, comparison operators next, and logical operators last. Comparison operators all have equal precedence; they are evaluated in the left-to-right order in which they appear. Arithmetic and logical operators are evaluated in the following order of precedence:

Arithmetic	**Comparison**	**Logical**
Negation (-)	Equality (=)	Not
Exponentiation (^)	Inequality (<>)	And
Multiplication and division (*, /)	Less than (<)	Or
Integer division (\)	Greater than (>)	Xor
Modulus arithmetic (Mod)	Less than or equal to (<=)	
Addition and Subtraction (+, -)	Greater than or equal to (>=)	
String concatenation (&)	Is	

If addition and subtraction, multiplication and division, occur together respectively in an expression, each operation is evaluated as it occurs from left to right.

The string concatenation operator (&) is not an arithmetic operator, but in precedence, it does fall after all arithmetic operators and before all comparison operators. The Is operator is an object reference comparison operator. It does not compare objects or their values; it checks only to determine whether two object references refer to the same object.

& Ampersand

Description

This operator forces text string concatenation of two expressions. Text concatenation operator connects or concatenates two values to produce a continuous text value.

Syntax

```
<expression> & <expression>
```

- The <expression> represents any valid logical expression.

Whenever an expression is not a string, it is converted to a String subtype. If both expressions are Null, the result is Null. However, if only one expression is Null, that expression is treated as a zero-length string ("") when concatenated with the other expression. Any expression that is Empty is also treated as a zero-length string.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego

DEFINE NameVar TEXTINPUT

ASSIGN NameVar=LastName&FirstName

LIST NameVar LastName FirstName
```

= Equal Sign

Description

This operator assigns a value to a variable or property. Comparison operator also used as an equal to; the result of comparison operators is usually a logical value, either true or false.

Syntax

<variable> <operator> <value>

- The <variable> represents any variable or any writable property.
- The <value> represents any numeric or string literal, constant, or expression.

Comments

The name on the left side of the equal sign can be a simple scalar variable or an element of an array. Properties on the left side of the equal sign can only be those writable properties at run time.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego

DEFINE Newvar NUMERIC

ASSIGN Newvar =Age

LIST Newvar Age
```

Addition (+)

Description

This operator provides the sums of two numbers. Basic arithmetic operator used for addition; the result of an arithmetic operator is usually a numeric value.

Syntax

```
[expression1] <operator> [expression2]
```

Comments

Although the + operator can be used to concatenate two character strings, the & operator should be used for concatenation to eliminate ambiguity and provide self-documenting code. If + operator is used, there may be no way to determine whether addition or string concatenation will occur. The underlying subtype of the expressions determines the behavior of the + operator in the following way:

If	Then
Both expressions are numeric	Add
Both expressions are strings	Concatenate
One expression is numeric and the other is a string	Add

If one or both expressions are Null expressions, the result is Null. If both expressions are Empty, the result is an integer subtype. However, if only one expression is Empty, the other expression is returned unchanged as a result.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego

DEFINE Newvar NUMERIC

ASSIGN Newvar = Age + 5

LIST Age Newvar
```

AND

Description

This operator performs logical conjunction on two Boolean expressions. If both expressions evaluate to True, the AND operator returns True. If either or both expressions evaluate to False, the AND operator returns False.

Syntax

[Logical Expression] AND [Logical Expression]

Comments

The expression is any valid logical expression in Epi Info.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Smoke

DEFINE Result TEXTINPUT

IF Age > 75 AND Sex = 2 THEN

    ASSIGN Result="Senior"

END
```

```
SELECT Result = "Senior"

LIST Result Age Sex
```

In this case, the value of "Senior" is assigned to all records that meet both criteria Age>65 and Sex=2.

ARITHMETIC

Description

These basic arithmetic operators can be used in combination with other commands. The result is a numeric value.

Syntax

```
[Expression] <Operator> [Expression]
```

- [Expression] is a numeric value or a variable containing data in numeric format.

Comments

The results are expressed in numeric format. The basic mathematical operators that can be used in Epi Info are as follows:

- **Addition + Basic** arithmetic operator used for addition; the result of an arithmetic operator is usually a numeric value (i.e., EX. 3 + 3).
- **Subtraction** - (Used for subtraction or negation.) Basic arithmetic operator used for subtraction or negation; the result of an arithmetic operator is usually a numeric value (i.e., EX. 3 – 1).
- **Multiplication * (Asterisk)** Basic arithmetic operator used for multiplication; the result of an arithmetic operator is usually a numeric value.
- **Division / Basic** arithmetic operator used for division; the result of an arithmetic operator is usually a numeric value.
- **Exponentiation ^**
- **Modulus or Remainder MOD**

Arithmetic operators are shown in descending order of precedence. Parentheses can be used to control the order in which operators are evaluated. The default order, however, frequently achieves the correct result.

While it is possible to do date math with dates considered as a number of days (e.g., IncubationDays = SymptomDateTime - ExposureDateTime), the behavior of the database services underlying Epi Info makes it more efficient to use time interval functions (e.g., IncubationDays = MINUTES(ExposureDateTime, Symptom DateTime)/[24*60]). For doing date math, the following rules apply:

Date + Date produces Date

Date - Date produces Days

Date * Date not permitted

Date / Date not permitted

Date ^ Date not permitted

Date + Number produces Date

Number + Date produces Number

The last two rules apply as well to other math operations: -, *, /, ^

The "zero day" for date math is December 30, 1899.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Surveillance
DEFINE var1 NUMERIC
ASSIGN var1=1250 MOD 100
DEFINE var2 NUMERIC
ASSIGN var2=1+1
DEFINE var3 NUMERIC
ASSIGN var3=2-1
DEFINE var4 NUMERIC
ASSIGN var4=1*1
DEFINE var5 NUMERIC
ASSIGN var5=8/4
DEFINE var6 NUMERIC
ASSIGN var6=5^2
LIST var1 var2 var3 var4 var5 var6
```

COMPARISONS

Description

These comparison operators can be used in If, Then, and Select statements in Check Code and Analysis programs. Yes/No variables can only be tested for equality against other Yes/No constants (+), (-), and (.).

Operator	Description
=	Equal to Comparison operator used for equal to; the result of comparison operators is usually a logical value, either True or False. EX. A1 = B1
>	Greater than comparison operator. Compares a value greater than another value; the result of comparison operators is usually a logical value, either True or False. Comparison operator used for comparing a value greater than another value; the result of comparison operators is usually a logical value, either True or False. EX. A1 > B1.
<	Less than comparison operator. Compares a value less than another value; the result of comparison operators is usually a logical value, either True or False. Comparison operator used for comparing a value less than another value; the result of comparison operators is usually a logical value, either True or False. EX. A1< B1
>=	Greater than or equal to

Operator	Description
<=	Less than or equal to
<>	Not equal to
LIKE	Left side variable matches right side pattern; in pattern, '*' matches any number of characters, '?' matches any one character.

Syntax

[Expression] <Operator> [Expression]

[Expression] is any valid expression.

Comments

Comparison operators are executed from left to right. There is no hierarchy of comparison operators. The <> operator can be used only with numeric variables. For non-numeric variables, use NOT.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Surveillance

SELECT Age>20

LIST Age Disease


READ {C:\My_Project_Folder\Sample\Sample.prj}:Surveillance

SELECT Age<45

LIST Age Disease


READ {C:\My_Project_Folder\Sample\Sample.prj}:Surveillance

SELECT Age>=38

LIST Age Disease


READ {C:\My_Project_Folder\Sample\Sample.prj}:Surveillance

SELECT Age<>77

LIST Age Disease
```

LIKE Operator

Description

This operator is used with the SELECT command to locate subsets of information using a wildcard search. LIKE can be used only to locate data in text variables and uses asterisks (*) to define the select value. It can also be used to create IF/THEN statements.

Syntax

SELECT <variable> LIKE "*value*"

SELECT <variable> LIKE "*val*"

SELECT <variable> LIKE "v*"

SELECT <variable> LIKE "*v"

- The select variable must be a text type. The value can be a whole or partial text value. Text variables must be enclosed in quotes.

Comments

The results appear in the Output window. Use LIST to view the selected records.

Examples

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Surveillance

DEFINE Sick NUMERIC

IF Disease LIKE "h*" THEN

    ASSIGN Sick = 0

END

SELECT Disease LIKE "h*"

LIST Age Disease DateAdmitted Sick GRIDTABLE
```

NOT

Description

This operator reverses the True or False value of the logical expression that follows.

Syntax

```
NOT [Expression]
```

The expression represents any valid logical expression in Epi Info.

Comments

If the value of an expression is True, Not returns the value False. If the expression is False, Not <expression> is True.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego

DEFINE NoVanilla YN
```

```
IF NOT Vanilla = (+) THEN
       NoVanilla = (+)
ELSE
   NoVanilla = (-)
END
FREQ NoVanilla Vanilla
```

VANILLA	NOVANILLA
Yes	No
No	Yes

OR

Description

This operator returns True if one or the other or both expressions are True. If either expression evaluates to True, Or returns True. If neither expression evaluates to True, Or returns False.

Syntax

[Logical Expression] OR [Logical Expression]

[Logical Expression] represents any valid logical expression in Epi Info.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego
DEFINE IceCream YN
IF VANILLA=(+) OR CHOCOLATE=(+) THEN
   IceCream=(+)
ELSE
  IceCream=(-)
END
FREQ IceCream
```

VANILLA	CHOCOLATE	ICE CREAM
Yes	Yes	Yes
No	Yes	Yes
Yes	No	Yes
No	No	No
Yes	Yes	Yes

XOR (eXclusive OR)

Description

This operator performs a logical exclusion on two expressions.

Syntax

```
[Logical Expression] XOR [Logical Expression]
```

The [Logical Expression] represents any valid logical expression in Epi Info 7 for Windows.

Comments

If one, and only one, of the expressions evaluates to True, the result is True. However, if either expression is Null, the result is also Null. When neither expression is Null, the result is determined according to the following table:

If expression1 is	**And expression2 is**	**Then result is**
True	True	False
True	False	True
False	True	True
False	False	False

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego

DEFINE Oneicecream YN

IF Vanilla = (+) XOR Chocolate = (+) THEN

   Oneicecream = (+)

ELSE

   Oneicecream = (-)

END

LIST Vanilla Chocolate Oneicecream GRIDTABLE
```

Functions

Do not put a space before the first parenthesis. Functions take the value of one or more variables and return the result of a calculation or transformation.

ABS Function

Description

The ABS function returns the absolute value of a variable by removing the negative sign, if any.

Syntax

```
ABS<variable>
```

- The <variable> can be an existing numeric variable, a defined variable containing numbers, or a numeric constant.

Comments

Results will be numeric.

Value	**ABS Function**
-2	2
1	1
0	0
-0.0025	0.0025

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego
DEFINE Age2 NUMERIC
DEFINE Age3 NUMERIC
ASSIGN Age2 = Age * -1
ASSIGN Age3 = ABS(Age2)
LIST Age Age2 Age3
```

DAY

Description

The DAY function extracts the day from the date.

Syntax

```
DAY (<variable>)
```

The <variable> is in date format.

Comments

If the date is stored in a text variable, the function will not be processed, and will be null.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego
DEFINE CurrentDay NUMERIC
ASSIGN CurrentDay = DAY(01/15/2007)
LIST CurrentDay
```

DAYS

Description

The DAYS function returns the number of days between <var2> and <var1>. If any of the variables or values included in the formula is not a date, the result will be null.

Syntax

```
DAYS(<var1>, <var2>)
```

The <variable> is in a date format.

Comments

If the date stored in <var1> is more recent than that in <var2>, the result is the difference in days expressed as a negative number.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego
DEFINE SickDays NUMERIC
ASSIGN SickDays = DAYS(04/18/1940, DateOnset)
LIST SickDays GRIDTABLE
```

EXISTS

Description

This function returns True if a file exists. Otherwise, it returns False.

Syntax

```
EXISTS(<variable>)
```

<variable> represents the complete file path and name in text format.

Comments

If you do not have permission to access the file, a False may be returned.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego
DEFINE var1 TEXTINPUT
ASSIGN var1="C:\epi_info\epimap.exe"
IF EXISTS(Var1) =(+) then
  DIALOG "Hello"
END
IF Exists("C:\Epi_Info\EpiInfo.mnu")=(+) then
     DIALOG "File epiInfo.mnu exists"
END
```

EXP

Description

This function raises the base of the natural logarithm (e) to the power specified.

Syntax

```
EXP(<variable>)
```

Comments

This variable can be an existing numeric variable, a defined variable containing numbers, or a numeric constant.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego
DEFINE ExpA NUMERIC
ASSIGN ExpA=EXP(Age)
LIST ExpA Age
```

FILEDATE

Description

This function returns the date a file was last modified or created. If FILEDATE is specified with a file path that lacks a directory, the current directory is used. If FILEDATE is specified without a file, or with a file that does not exist, the function returns missing.

Syntax

```
FILEDATE(<variable>)
```

The <variable> represents the complete file path and the name is text format.

Comments

This function is useful when several users are updating a large database.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:RHepatitis

DEFINE NewUpdate DATEFORMAT

ASSIGN NewUpdate=FILEDATE("C:\epi_info\Sample.mdb")

IF FILEDATE("C:\epi_info\Sample.mdb") > NewUpdate THEN

   DIALOG "This information may be out of date. Please check the source."
   TITLETEXT="Warning"

END

LIST NewUpdate
```

FINDTEXT

Description

This function returns the position in a variable in which the string is located.

Syntax

```
FINDTEXT(<variable1>,<variable2>)
```

The <variable1> represents the string of characters to be found. The <variable2> represents the string to be searched.

Comments

If the sting is not found, the result is 0; otherwise it is a number corresponding to the position of the string starting from the left. The first character is 1. If the result is 0, the test was not found.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Surveillance

DEFINE Var11 NUMERIC
```

```
VAR11=FINDTEXT("M",LASTNAME)
LIST LASTNAME Var11
```

FORMAT

Description

This function changes the format of one variable type to text in a specified format. If no format is specified it returns text and converts a number to text.

Syntax

```
FORMAT(<variable>,["Format Specification"])
```

The <variable> represents a variable in any format and the [Format Specification] can represent any of the following:

Format Specification	**Description**
Date Formats	
General Date	11/11/1999 05:34
Long Date	System's long date format
Medium Date	System's medium date format
Short Date	System's short date format
Long Time	System's long time format
Medium Time	System's medium time format
Short Time	System's short time format
Number Formats	
General Number	No thousand separator
Currency	Thousand separator plus two decimal places (based on system settings)
Fixed	At least #.##
Standard	#,###.##
Percent	Number multiplied by 100 plus a percent sign
Scientific	Standard scientific notation

Format Specification	**Description**
Yes/No	Displays NO if number = 0, else displays Yes
True/False	False if number = 0
On/Off	True if number <> 0 Displays 0 if number = 0, else displays 1
Custom Format	Allows for the creation of customized formats

Comments

```
Output may vary based on the specific configuration settings of the local computer.

Format(Time, "Long Time")

MyStr = Format(Date,"Long Date")

MyStr = Format(MyTime,"h:m:s")

Returns "17:4:23"

MyStr = Format(MyTime,"hh:mm:ssAMPM")

Returns "05:04:23 PM"

MyStr = Format(MyDate,"dddd, mmm yyyy")

Returns "Wednesday, ' Jan 27 1993". If format is not supplied, a string is returned.

MyStr = Format(23)

Returns "23".


User-defined formats

MyStr = Format(5459.4, "##,##0.00")

Returns "5,459.40"

MyStr = Format(334.9, "###0.00")

Returns "334.90"

MyStr = Format(5, "0.00%")

Returns "500.00%"

MyStr = Format("HELLO", "<")

Returns "hello"

MyStr = Format("This is it", ">")

Returns "THIS IS IT"

MyStr = Format("This is it", ">;*")

Returns "THIS IS IT"
```

Examples

```
READ 'C:\Epi_Info\Refugee.MDB':Patient
DEFINE var2 NUMERIC
DEFINE var3 NUMERIC
DEFINE var4 NUMERIC
DEFINE var5 NUMERIC
DEFINE var6 NUMERIC
DEFINE var7 YN
DEFINE var8 Boolean
DEFINE var9
DEFINE var10
var2=FORMAT(BOH, "Currency")
var3=FORMAT(BOH, "fixed")
var4=FORMAT(BOH, "Standard")
var5=FORMAT(BOH, "Percent")
var6=FORMAT(BOH, "Scientific")
var7=FORMAT(BOH, "Yes/No")
var8=FORMAT(BOH, "True/false")
var9=FORMAT(BOH, "On/Off")
var10=FORMAT(BOH, "VB\s #,###.##")
LIST dob var2 var3 var4 var5 var6 var7 var8 var9 var10
```

HOUR

Description

This function returns a numeric value that corresponds to the hour recorded in a date/time or time variable.

Syntax

```
HOUR(<variable>)
```

The <variable> represents a variable in date format.

Comments

If the time is stored in a text variable, the function will not be processed, and the result will be null.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego
DEFINE Local DATEFORMAT
ASSIGN Local = SYSTEMTIME
```

```
LIST Local

DEFINE hour1 NUMERIC

ASSIGN hour1=hour(local)

LIST Local hour1
```

HOURS

Description

This function returns the number of hours between <var1> and <var2> in numeric format.

Syntax

```
HOURS(<var1>, <var2>)
```

<var1> and <var2> represent variables in time or date/time format.

Comments

If the date stored in <var1> is older than that in <var2>, the result will be the difference in hours expressed as a negative number. Both variables must contain data in date, time, or date/time format. If any of the variables or values included in the formula is not a date, the result will be null.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego

DEFINE hour1 NUMERIC

ASSIGN hour1=HOURS(Timesupper,Dateonset)

LIST hour1

LIST hour1 Timesupper Dateonset
```

LN

Description

The function LN returns the natural logarithm (logarithm in base e) of a numeric value or variable. If the value is zero or null, it returns a null value.

Syntax

```
LN(<variable>)
```

The <variable> can be an existing numeric variable, a defined variable containing numbers, or a numeric constant.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego

DEFINE Natlogofage NUMERIC
```

```
ASSIGN Natlogofage = LN(AGE)
LIST Age Natlogofage
```

LOG

Description

This function returns the base 10 logarithm (decimal logarithm) of a numeric value or variable. If the value is 0 or null it returns a null value.

Syntax

```
LOG(<variable>)
```

The <variable> can be an existing numeric variable, a defined variable containing numbers, or a numeric constant.

Comments

The results will be numeric.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego
DEFINE Declog NUMERIC
ASSIGN Declog = LOG(Age)
LIST Age Declog
```

MINUTES

Description

This function returns the number of minutes between <var1> and <var2> in numeric format.

Syntax

```
MINUTES(<var1>, <var2>)
```

<var1> and <var2> represent variables in time or date/time format.

Comments

If the date stored in <var1> is older the one in <var2>, the result will be the difference in minutes expressed as a negative number. Both variables must contain data in date, time, or date/time format. If any of the variables or values included in the formula is not a date, the result will be null.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego

DEFINE Min1 NUMERIC

ASSIGN Min1=MINUTES(timesupper,dateonset)

LIST Min1
```

MONTH

Description

This function extracts the month from the date.

Syntax

```
MONTH(<variable>)
```

The <variable> represents a variable in date format.

Comments

If the date is stored in a text variable, the function will not be processed, and the result will be null.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego

DEFINE CurrMonth NUMERIC

ASSIGN CurrMonth = MONTH(01/01/2005)

LIST CurrMonth
```

MONTHS

Description

This function returns the number of months between <var1> and <var2>. If any of the variables or values included in the formula is not a date, the result will be null.

Syntax

```
MONTHS(<var1>, <var2>)
```

<var1> and <var2> represent variables in date format.

Comments

If the date stored in <var1> is older than that in <var2>, the result will be the difference in months expressed as a negative number.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Surveillance

DEFINE AgeMonths NUMERIC

ASSIGN AgeMonths = MONTHS(BirthDate,01/01/2000)

LIST AgeMonths
```

NUMTODATE

Description

This function transforms three numbers into a date format.

Syntax

```
NUMTODATE(<year>, <month>, <day>)
```

- <year> represents a numeric variable or a number representing the year.
- <month> represents a numeric variable or a number representing the month.
- <day> represents a numeric variable or a number representing the day.

Comments

If the date resulting from the conversion is not valid (e.g., December 41, 2000), the date is recalculated to the corresponding valid value (e.g., January 10, 2001). When <Year> ranges between 0 and 29, it is represented as the respective year between 2000 and 2029. Values from 30 to 99 are represented as the respective year between 1930 and 1999. The earliest date that can be recorded is Jan 01, 100.

Day	Month	Year	Date Created
02	02	1999	02/02/1999
60	01	1999	03/01/1999
15	18	2000	03/18/2001
99	99	99	06/07/0107
20	74	74	08/20/1974

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Surveillance
DEFINE day1 NUMERIC
DEFINE month1 NUMERIC
DEFINE year1 NUMERIC
ASSIGN day1= day(BirthDate)
ASSIGN month1 = month(BirthDate)
ASSIGN year1 = year(BirthDate)
define date2 DATEFORMAT
ASSIGN date2= NUMTODATE(year1,month1,day1)
LIST month1 day1 year1 date2 BirthDate GRIDTABLE
```

NUMTOTIME

Description

This function transforms three numbers into a time or date/time format.

Syntax

```
NUMTOTIME(<hour>, <minute>, <second>)
```

- <hour> represents a numeric constant or variable representing hours.
- <minute> represents a numeric constant or variable representing minutes.
- <second> represents a numeric constant or variable representing seconds.

Comments

Time must be entered in 24-hour format. Invalid dates will be recalculated to the respective valid time. If the number of the hour exceeds 24, the resulting variable will have a date/time format and the default day 1 will be December 31, 1899.

Hour	Minute	Second	Time Created
00	00	00	12:00:00 AM
00	00	90	12:01:30 AM
15	84	126	04:26:06 PM
25	00	00	12/31/1899 1:00:00 AM
150	250	305	01/05/1900 10:15:05 AM
15999	7500	8954	09/21/1901 07:29:14 AM

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Surveillance
DEFINE Var3 DATEFORMAT
ASSIGN Var3=SYSTEMTIME
DEFINE Hour1 NUMERIC
DEFINE Minute1 NUMERIC
DEFINE Second1 NUMERIC
ASSIGN Hour1=HOUR(VAR3)
ASSIGN Minute1=MINUTE(VAR3)
ASSIGN Second1=SECOND(VAR3)
DEFINE Time2 DATEFORMAT
ASSIGN Time2=NUMTOTIME(HOUR1,MINUTE1,SECOND1)
LIST Var3 Hour1 Minute1 Second1 Time2
```

RECORDCOUNT

Description

This function returns the number of records in the current View. In Analysis, this takes into account any SELECT statement and value of the Process (Deleted) setting.

Syntax

```
RECORDCOUNT
```

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego
IF RECORDCOUNT=0 THEN
   DIALOG "No records found."
   QUIT
END
```

RND

Description

This function generates a random number between <var1> and <var2>.

Syntax

RND(<min>, <max>)

- The <min> represents a number or numeric variable that corresponds to the lowest value of the random number to be generated.
- The <max> represents a number or numeric variable that corresponds to the highest possible value for the random number to be generated.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego
DEFINE Random1 NUMERIC
DEFINE Random2 NUMERIC
DEFINE Random3 NUMERIC
ASSIGN Random1=RND(1,100)
ASSIGN Random2=RND(1,100)
ASSIGN Random3=RND(1,100)
LIST Random1 Random2 Random3
```

ROUND

Description

This function rounds the number stored in the variable to the closest integer. Positive numbers are rounded up to the next higher integer if the fractional part is greater than or equal to 0.5. Negative numbers are rounded down to the next lower integer if the fractional part is greater than or equal to 0.5.

Syntax

```
ROUND(<variable>)
```

The <variable> can be an existing numeric variable, a defined variable containing numbers, or a numeric constant.

Comments

The results are returned in numeric format.

Differences Between TRUNC and ROUND		
Value	**TRUNC**	**ROUND**
0.123456	0	0
7.99999999	7	8
45.545	45	46

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego

FREQ AGE

DEFINE Decade NUMERIC

ASSIGN Decade=ROUND(AGE/10)+1

LIST AGE Decade
```

SECONDS

Description

This function returns the number of seconds between <var1> and <var2> in numeric format.

Syntax

```
SECONDS(<var1>, <var2>)
```

<var1> and <var2> represent variables in time or date/time format.

Comments

If the date stored in <var1> is older than that in <var2>, the result will be the difference in seconds expressed as a negative number. Both variables must contain data in date, time or date/time format. If any of the variables or values included in the formula is not a date, the result is null.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego

DEFINE Sec1 NUMERIC

ASSIGN Sec1=SECONDS(Timesupper,DateOnset)
```

```
LIST Timesupper DateOnset Sec1
```

SIN, COS, TAN

Description

These functions return the respective trigonometric value for the specified variable.

Syntax

```
SIN(<variable>)
```

The <variable> can be an existing numeric variable, a defined variable containing numbers, or a numeric constant.

Comments

The variable is interpreted as the angle in radians. To convert degrees to radians, multiply by pi (3.1415926535897932) divided by 180.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego


DEFINE SinA NUMERIC

DEFINE SinB NUMERIC

DEFINE CosA NUMERIC

DEFINE TanA NUMERIC

ASSIGN SinA=SIN(AGE)

ASSIGN SinB=SIN(AGE)*3.14/180

ASSIGN CosA=COS(AGE)

ASSIGN TanA=TAN(AGE)

LIST SinA CosA TanA SinB
```

SUBSTRING

1.1.1.2 Description

This function returns a string that is a specified part of the value in the string parameter.

Syntax

```
SUBSTRING(<variable>, [First], [Length])
```

- The <variable> represents a variable in text format.

- The [First] represents the position of the first character to extract from the file.
- The [Length] represents the number of characters to extract.

Comments

This function cannot be used with non-string variables.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego
DEFINE Text1 TEXTINPUT
ASSIGN Text1 ="James Smith"
DEFINE LName TEXTINPUT
ASSIGN LName = SUBSTRING(Text1,7,5)
LIST Text1 LName
```

SYSTEMDATE

Description

This function returns the date stored in the computer's clock.

Syntax

```
SYSTEMDATE
```

Comments

The SYSTEMDATE cannot be changed (assigned) from Classic Analysis. To use the SYSTEMDATE for computations, a new variable must be defined.

1.1.1.3 Example

To calculate next week's date:

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Surveillance
DEFINE TodayDate DATEFORMAT
ASSIGN TodayDate =SYSTEMDATE + 7
LIST TodayDate
```

SYSTEMTIME

Description

This function returns the time stored in the computer's clock at the time the command is executed.

Syntax

```
SYSTEMTIME
```

Comments

The SYSTEMTIME cannot be changed from Classic Analysis (assigned). To use the system time for computations, a new variable must be defined.

Example

To calculate a time two hours after the current time:

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Surveillance
DEFINE Later DATEFORMAT
ASSIGN Later =SYSTEMTIME
LIST Later
ASSIGN Later =SYSTEMTIME+(120)
LIST Later
```

TRUNC

Description

This function removes decimals from a numeric variable, returning the integer part of the number. This follows the same logic as rounding toward zero.

Syntax

TRUNC(<variable>)

The <variable> can be an existing numeric variable, a defined variable containing numbers, or a numeric constant.

Comments

The result will be returned in numeric format.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:ADDFull
DEFINE Trc1 Numeric
ASSIGN Trc1 = TRUNC(ADDSC)
1.1.2  LIST Trc1 ADDSC
```

TXTTODATE

Description

This function returns a date value that corresponds to the string.

Syntax

```
TXTTODATE(<variable>)
```

The <variable> represents a variable in text format.

Comments

The text variable can be in any format that can be recognized as a date (e.g., "Jan 1, 2000", "1/1/2000").

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Surveillance
DEFINE Var1 TEXTINPUT
ASSIGN Var1="05/20/2006"
DEFINE Var2 DATEFORMAT
ASSIGN Var2=TXTTODATE(Var1)
DISPLAY DBVARIABLES
LIST Var1 Var2
```

TXTTONUM

Description

This function returns a numeric value that corresponds to the string.

Syntax

```
TXTTONUM(<variable>)
```

The <variable> represents a variable in text format.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Oswego
DEFINE Var1 TEXTINPUT
ASSIGN Var1="12345"
DEFINE Var2 NUMERIC
ASSIGN Var2=TXTTONUM(Var1)
LIST Var1 Var2
DISPLAY DBVARIABLES
```

UPPERCASE

Description

This function returns a string (text) variable that has been converted to uppercase.

Syntax

```
UPPERCASE(<variable>)
```

The <variable> represents a variable in text format.

Comments

Only lowercase letters are converted to uppercase; all uppercase letters and non-letter characters remain unchanged.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Surveillance
DEFINE LastName2 TEXTINPUT
ASSIGN LastName2 = UPPERCASE(LASTNAME)
LIST LastName2 LASTNAME
```

YEAR

Description

This function extracts the year from a date.

Syntax

```
YEAR(<variable>)
```

The <variable> represents a variable in date format.

Comments

The date argument is any expression that can represent a date. If the date variable contains null, null is returned.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Surveillance
DEFINE CurrentYear NUMERIC
ASSIGN CurrentYear =YEAR(01/01/2006)
LIST CurrentYear
```

YEARS

Description

This function returns the number of years from <var1> to <var2> in numeric format. If any of the variables or values included in the formula is not a date, the result will be null.

Syntax

```
YEARS(<var1>, <var2>)
```

<var1> and <var2> are represented in date format.

Comments

If the date stored in <var1> is more recent than that in <var2>, the result will be the difference in years expressed as a negative number.

Example

```
READ {C:\My_Project_Folder\Sample\Sample.prj}:Surveillance
DEFINE SurveyDate DATEFORMAT
ASSIGN SurveyDate=05/15/2001
DEFINE AgeYears NUMERIC
ASSIGN AgeYears =YEARS(BirthDate,SurveyDate)
MEANS AgeYears
LIST AgeYears BirthDate SurveyDate
```

15. Glossary

#

.CHK: EpiData was developed for data entry as an update of the principles used in the DOSprogram Epi Info v6. It is an all-in-one program (one exe file) for Windows (95/98/NT/2000) and Macintosh (with RealPc emulation). EpiData uses Epi Info v6 format for files (Qes, Rec, and Chk). Data can be exported to CSV, (comma separated data), dBase, Exceland Stata v4-6. Simple (range, legal, date) and enhanced control of logical consistency across variables, jumps based on the value of entry, and calculations during data entry is easy to define. Lists of data and overall frequency tables can be produced. Compare two files and get a list of differences in data (validate).

.QES: EpiData was developed for data entry as an update of the principles used in the DOSprogram Epi Info v6. It is an all-in-one program (one exe file) for Windows (95/98/NT/2000) and Macintosh (with RealPc emulation). EpiData uses Epi Info v6 format for files(Qes, Rec, and Chk). Data can be exported to CSV, (comma separated data), dBase, Exceland Stata v4-6. Simple (range, legal, date) and enhanced control of logical consistency across variables, jumps based on the value of entry, and calculations during data entry is easy to define. Lists of data and overall frequency tables can be produced. Compare two files and get a list of differences in data (validate).

.REC: EpiData was developed for data entry as an update of the principles used in the DOSprogram Epi Info v6. It is an all-in-one program (one exe file) for Windows (95/98/NT/2000) and Macintosh (with RealPc emulation). EpiData uses Epi Info v6 format for files (Qes, Rec and Chk). Data can be exported to CSV, (comma separated data), dBase, Exceland Stata v4-6. Simple (range, legal, date) and enhanced control of logical consistency across variables, jumps based on the value of entry, and calculations during data entry is easy to define. Lists of data and overall frequency tables can be produced. Compare two files and get a list of differences in data (validate).

95% Confidence Limits: A range of values for a variable that indicates the likely location of the true value of a measure.

A

Association: Statistical relationship between two or more events, characteristics, or other variables.

Average: Average of the values in the numeric expression.

C

Case: In epidemiology, a countable instance in the population or study group of a particular disease, health disorder, or condition under investigation. Sometimes, an individual with the particular disease.

Case Based: An advanced display process that allows users to show different symbols based on levels of classification (e.g., Confirmed, Probable, Discarded, Suspected).

Chi Square: A test of statistical significance used to determine how likely an observed association between an exposure and a disease could have occurred because of chance alone, if the exposure was not actually related to the disease. Tests for the presence of a trend in dose response or other case control studies where a series of increasing or decreasing exposures is being studied.

Conditional Probability: Estimate of the probability of survival of a defined group at a designated time interval.

Control: In a case-control study, comparison group of persons without disease.

Count: Number of values in the expression or number of selected rows.

D

Date Literals: A specific date used in functions or commands in Check Code or in Classic Analysis.

DLL: Dynamic Link Library

E

Elementary outcomes: All possible results of a random experiment.

Epi Info: Epi Info for Windows

Epidemic: The occurrence of more cases of disease than expected in a given area or among a specific group of people over a particular period of time.

Epidemiology: The study of the distribution and determinants of health-related states or events in specified populations, and the application of this study to the control of health problems.

Evaluation: A process that attempts to determine as systematically and objectively as possible the relevance, effectiveness, and impact of activities in the light of their objectives.

F

Frequency: The count in any given interval. The relative frequency is the proportion of weights in each interval.

G

GIS: Geographic Information System

H

Histogram: A graphical representation of the frequency of data values within small data ranges which are created by dividing the total range of data into 5 to 20 equal subintervals. The most common histogram form is the Bell Curve, also known as a Normal Distribution.

I

Interaction: The odds ratio (OR) for a variable varies with the value of another variable.

M

Maximum: Highest value in the expression.

Mean: Equal to the average of the data. Add all data together and divide by the number of observations.

Median: The measure of central location which divides a set of data into two equal parts. A center or mid-point of the data. Order the data from the smallest to the largest and the center point is the median. For odd numbers, it is the center number. For even numbers, it is the average of the center numbers.

Minimum: Lowest value in the expression.

O

ODBC: Open Database Connectivity

Odds Ratio: A measure of association that quantifies the relationship between an exposure and health outcome from a comparative study. Also known as the cross-product ratio.

Outbreak: Synonymous with epidemic. Sometimes the preferred word because it may escape sensationalism. Alternatively, a localized as opposed to generalized epidemic.

Outliers: The number of data values that are much smaller or larger than the rest of the data values.

P

P-Value: The probability that an observed association between an exposure and a disease could have occurred because of chance alone, if the exposure was not actually related to the disease.

PGM Files: Commands entered into Classic Analysis generate lines of code in the Program Editor and can be stored in the current PRJ file. Programs can be saved internally within the project or externally as a text file with a .pgm7 file extension. This makes a neat package of data and programs that can be copied to another system or sent by email for use elsewhere.

R

Random experiment: The process of observing the outcome of a chance event.

Ratio: The value obtained by dividing one quantity by another.

Risk: The probability that an event will occur (e.g., an individual will become ill or die within a stated period of time or age).

Risk Ratio: A comparison of the risk of some health-related event (e.g., disease or death in two groups).

S

Sample space: The set or collection of all elementary outcomes.

Standard Deviation: A statistical summary of how dispersed the values of a variable are around its mean. The average of all distances of each data point from the mean.

Standard Error: (of the mean) The standard deviation of a theoretical distribution of sample means of a variable around the true population mean of that variable.

Strata: A population subgroup defined by any number of demographic characteristics (e.g., age, gender, race, etc.).

Sum: Total of the values in the numeric expression.

V

Variable: Any characteristic or attribute that can be measured.

Variance: A measure of the dispersion shown by a set of observations.

16. Appendix

Data Quality Check

Date Validation

The **Date** field is an alphanumeric field with pre-set date patterns selected from the pattern drop-down list. It cannot be altered.

The date field offers Range property, which can be applied to Date variable types. Range allows for a specified value between one setting and another. Values falling outside a specified range will prompt you with a warning message in the Enter module. Unless the field is designated Required, missing values are accepted.

How To:

- Use Form Designer to open the data entry form.
- Select **Oswego** from "C:\Epi Info 7\Projects\Sample\Sample.prj\Oswego".
- Right click on the **Date Onset** field.
- Select **Properties** option from the popup menu.
- Click on **Range** option.
- Enter the **Lower** and **Upper** date.

Numeric Data Validation – Lower and Upper Bound

The **Number** field is a numeric field that has six predefined value patterns (e.g., xxx.xx). A new pattern can be created by simply typing the pattern into the Pattern field.

The Number field offers Range property. The Range property can be applied to Number or Numeric types. The Range allows for a specified value between one setting and another. Values falling outside a specified range will prompt the user with a warning message in the Enter module. Missing values are accepted unless the field is also designated Required.

How to:

Use Form Designer to open the data entry form:

- Select **Oswego** from "C:\Epi Info 7\Projects\Sample\Sample.prj\Oswego".
- Right click on the **Age** field.
- Select **Properties** option from the popup menu.

- Click on **Range** option.
- Enter the **Lower** and **Upper** values.

Legal Values

A Legal Values field is a drop-down list of choices on the questionnaire. These items cannot be altered by the person entering data. The only values legal for entry are the ones in the list.

How to:

- See the **Legal Value** section

Comment Legal Values

Comment Legal variables are similar to Legal Values. Comment Legal fields are text fields with a code typed in front of text (with a hyphen) so that in populating fields, the code is entered instead of the text. In the Classic Analysis module, the statistics are displayed with the codes only.

How To

- See the Comment Legal section

Auto Search

During data entry, fields with Autosearch Check Code are automatically searched for one or more matching records. If found, a match can be displayed and edited or be ignored and data entry can continue on the current record. Autosearch is used as an alert to potential duplicate records. However, the Autosearch command does not restrict users from entering duplicate records.

How To

- See the AutoSearch section

Skip Logic/Patterns

Skip patterns can be created by changing the tab order and setting a new cursor sequence through a questionnaire, or by creating Check Code using the GOTO command. Skip patterns can also be created based on the answers to questions using an IF/THEN statement.

How To

- See the Skip section.

Update Data

This technique allows users to search for existing records and re-enter or update data.

How To

- See the Data Entry sections.

Must Enter

A Must Enter/Required field is a mandatory field. Since the properties are mutually exclusive, Required cannot be used in combination with Read Only. If a page contains a Required variable, the Enter module will prevent further page navigation until a value has been entered. Use this property sparingly to avoid gridlock.

How To

- See the Must Enter section.

Calculated Age

If possible, always use the Years command to calculate from the birth and the data entry date.

How To

- See the YEARS function

Analysis

Check for Duplicate Records

Using Frequency, check for duplicate records.

How to:

- READ {C:\Epi Info 7\Projects\Sample\Sample.prj}:Oswego
- FREQ CODE
- SELECT CODE = "P55"
- LIST * GRIDTABLE

Delete Duplicate Records

- Helps you delete or remove duplicate records.

How To:

- READ {C:\Epi Info 7\Projects\Sample\Sample.prj}:Oswego
- FREQ CODE
- SELECT CODE = "P55"
- LIST * GRIDTABLE
- DELETE UNIQUEKEY = 1 PERMANENT

Missing Data

Using the SELECT command, this technique allows you to find missing records for specific fields or variables.

How To:

- READ {C:\Epi Info 7\Projects\Sample\Sample.prj}:Oswego
- SELECT DATEONSET = (.)
- LIST * GRIDTABLE