

Introduction

Clarion for Windows allows new components to be "plugged in" and used along with Clarion's standard templates--not only new templates, but also whole new classes of templates. ToolCraft Development Corporation has created the Power Templates product line, which is a whole new class of templates designed to provide more advanced features than those included in Clarion's standard templates.

Building advanced power into your Clarion for Windows applications is easy if you start with the right templates. ToolCraft is one of the leading third party vendors of Clarion for Windows add-on templates. Our templates help you create feature rich applications effortlessly.

Two of these templates, Power Browse and Power Popup, are included on this CD as TrialPak versions. A third Power Templates product, Chart Wrapper, is included in the demo application and discussed later in this chapter. The TrialPak versions have not been "crippled" in any way other than the warning message added to the executable. This gives you the ability to try out fully functioning copies of these products to see how easy it is to add advanced features to your applications.

The on-line help for both products contains all of the information that is found in the printed manuals normally included with the products, except for the tutorial chapters. This document contains portions of the tutorial chapters included with Power Browse and Power Popup.

You may also find these tutorials helpful in getting you started with the Clarion for Windows product itself. The Power Browse tutorial provides step-by-step instructions for building an application from the ground up, including creating a dictionary.

Documentation Standards

- **Bullets like this are used to indicate that you need to perform a set of steps to complete the process being discussed. These bullets are used in conjunction with the following:**
 1. Numbers like this one are used to list the steps that you should take to complete the particular process being discussed.

Italics are used to designate words or characters that you should actually type at the keyboard.

SMALL CAPS indicate keys to press at the keyboard or mouse controls such as RIGHT-CLICK.

Boldface type is used to designate field names, commands and options that you should select from a list or menu when completing a particular process. It is also used to indicate window or dialog box names (titles).

A single letter in a command is **underlined** to indicate that the letter it is the "hot" key available for the command being discussed. You can press ALT and the underlined key to complete the process instead of the instruction given. For example, if a command is given as follows: Press the **New** button, you can press ALT+N instead.

Exploring the Power Templates Demo

The demo application included with the TrialPak version of Power Templates gives a good overview of what kinds of additional functionality can be added to your own applications. This section will walk you through the example and explain the basic functions used on each of the screens.

The demo program consists of three main menu items: Power Browse, Power Popup & Chart Wrapper.

► To run the example application

1. From the **Windows Program Manager**, choose the **F**ile menu and then choose **R**un.
2. In the **Run** dialog, type `D:\3RDPARTY\TOOLS\TOOLCRFT\DEMO\SETUP` and press the **OK** button. This will load the example application onto your system and then run it.

The Power Browse Demo

► To view the topics covered in the Beginning Browse

1. Once the example application's menu appears, select the **P**ower **B**rowse menu option.
2. Select the Beginning item from the Power Browse menu option. The Beginning Browse Example window shown below will appear.
3. Notice the Recipe# and Name fields located below the browse list box--these are hot display fields. Press the DOWN-ARROW key several times and watch these fields automatically update each time you select a new entry in the list.
4. To test the Step search, type the letter C. The list will advance to **Cheesecake**, the first record found that begins with the letter C. Now, TAB to the **S**earch field, type *Onion Soup* and press TAB. This is an example of an Entry type search. As you can see, Power Browse supports the use of **BOTH** of them together on the same browse list box.
5. To resize the window, start by dragging the bottom of the window down. The browse list expands down to fill the length of the screen. Now click the maximize button in the upper right corner of the window. Notice again, that the browse list expands to fill the available space.
6. Press the **C**lose button to return to the menu.

► To view the topics covered in the Intermediate Browse

1. Select the Intermediate item from the Power Browse menu option. The Intermediate Browse Example window will appear.
2. A parent-child relationship exists between both of the browse lists on this window. Range limits have been defined so that only those child records associated with the current parent record will be displayed. Press the DOWN-ARROW key several times in the RECIPE list box (parent) and watch the records in the INGREDIENT list box (child) automatically change each time you select a new entry.
3. To test the resizing of both the parent and child browse list box, drag the bottom of the window down. Notice that both browse lists expand down to fill the length of the screen.
4. Try out the multiple display order control by selecting **Beef Dishes** only in the **S**ort **B**y field above the RECIPE browse list box. The filtering option was defined so that only those recipes with the word BEEF in their name would be displayed.

5. The mark buttons allow you to mark or unmark either one entry at a time or all of them at once. Try out each of these buttons. The message used to mark the records is user- definable so you could use any text that you want.
6. The following types of options are used on both browse list boxes: hot display fields, multiple display orders and search entry. Power Browse will support the use of all of the available options on each browse in the window.
7. Press the **Close** button to return to the menu.

► **To view the topics covered in the Advanced Browse**

1. Select the Advanced item from the Power Browse menu option. The Advanced Browse Example window shown below will appear.
2. Only one set of Update buttons is used for all four browse list boxes on this window. Select the RECIPE list box in the top left corner and press the **Change** button. Notice the Update Recipe form is displayed. Now select the MENU list box in the bottom right corner and press the **Change** button again. The Update Menu form is now displayed.
3. A parent and child relationship exists between the RECIPE browse list box and both the INGREDIENT and INSTRUCTION browse lists. Range limits have been defined so that only those child records associated with the current parent record will be displayed. Press the DOWN-ARROW key several times in the RECIPE list box (parent) and watch the records in the INGREDIENT and INSTRUCTIONS list boxes automatically change each time you select a new entry.
4. The RECIPE and INSTRUCTIONS browse list boxes have been defined as the parent and child browse list boxes to resize. Drag the bottom of the window down and notice that they both resize and the other two browse list boxes still remain intact.
5. The following types of options are used on several of the browse list boxes: hot display fields, multiple display orders, search entry and mark buttons. Again, Power Browse will support the use of the same options being used more than once.
6. Press the **Close** button to return to the menu.
7. Choose the **Exit** menu and then choose **Exit** to close the example application.

The Power Popup Demo

► **To view the popup menus in the example application**

1. Select the **Power Popup Test** item from the **Power Popup** menu option. When the browse window opens, RIGHT-CLICK in the browse box control.
2. Notice that some of the items have arrows next to them indicating that they have cascading (sub) menus. You can have an unlimited number of submenus under each menu.
3. The **Update Records** submenu items post the Accepted event back to the update button controls (which are hidden).
4. Select the **Change** record update item from the menu and then RIGHT-CLICK on the Update form. Notice that you can RIGHT-CLICK anywhere on the form and the same popup menu appears. The AliasPopup feature is used so that the menu is created only once and then associated to all of the other controls on the window.

5. Notice that the **Change Filters** submenu items use a check mark to indicate which filter is currently being viewed. These items change the local filter variable and then post the ScrollTop event to the list box to make the records redisplay.
6. The **View Ingredients** menu item calls another browse procedure with a parameter while the **Print Record** item calls another procedure with no parameters.
7. Notice the separator line between the **View Ingredients** and **Print Records** menu items.
8. Press the **Close** button to return to the menu.

The Chart Wrapper Demo

► To view the Chart Wrapper demo application

1. Select the **Open Graph** item from the **Graphing** menu option. When the **Open Graph** dialog opens, select the \CW\TCTPL\EXAMPLES\SAMPLES.GPH file using the ellipsis button. Press **OK** button **Select Graph File** dialog.
2. Notice that a list of available graphs are now displayed. You can add as many graphs to one file as you like. Select Bar Graph #1 and press **OK** in the **Open Graph** dialog.
3. Now that a graph is open, select the **Graphing** menu item again. Notice all of the additional items now available on this menu. Select the **Graph Settings** option to open the Tool Box. The Tool Box allows you to change all of the properties of the GRAPH.VBX control. You should read the documentation that came with the GRAPH.VBX control to understand what each of these properties do.
4. Start by changing the graph types and watch the graph redisplay. There are many different graph types. The documentation that came with the GRAPH.VBX control explains which ones allow multiple sets of data, etc. For example, pie charts will display only one set of data.
5. Select Data from the list of options. When entering data, notice that there is a column running down the side of the data values and a row running across the top of the data. The values on the top are used as labels for each point. These labels will run across the bottom of the graph. The values entered along the side will be used as the legend text. To delete the selected row of data press CTRL+R. To delete the selected column of data press CTRL+C.
6. Select Pattern from the list of options. To select a pattern for each element, you can either use the hot hit method (if you are not using version 2.0 of the GRAPH.VBX control) where you simply click on the actual GRAPH.VBX control or you can press the element number button to select a new element. When using multiple sets of data, this element button selects each set, otherwise it selects each point. These element buttons are used throughout the Tool Box and function in the same way in each location.
7. Press the **Close** button to return to the menu.

Now that you have completed the tour of the demo application, you should have a better idea of what types of features you can add to your own application by using Power Templates. Chapters 2 and 3 give instructions on how to recreate parts of this demo application. Before completing the exercises, you must follow the setup instructions in this chapter to load the Power Browse and Power Popup products onto your computer.

Setup

An install program is included on the CD which will copy the Power Browse and Power Popup files to your hard drive. Before you begin, please be aware that the TrialPak versions of these products are intended for evaluation purposes only. Applications created with these products may not be distributed without purchasing a license to use them.

The following instructions assume you are installing from drive D:. If you are not, modify the commands below to represent the location of the Power Templates install disk.

► **To Run the setup program**

1. Insert the TrialPak CD in drive D:.
2. From the **Windows Program Manager**, choose the **F**ile menu and then choose **R**un.
3. In the **Run** dialog, type `D:\3RDPARTY\TOOLS\TOOLCRFT\SETUP` and press the **OK** button.
4. Enter the drive where the TrialPak CD is located, the directory where you installed Clarion for Windows and the directory where you want to install Power Templates (if they are different from the defaults). Press the **Install** button to begin the installation.
5. After the files are copied and the setup program is finished, read the README file and then follow the instructions below to register the templates.

Although all of the Power Browse and Power Popup files are now located on your system, the ToolCraft Class of templates must be registered within the Clarion template registry before they can be used.

► **To add the ToolCraft class to the Clarion template registry**

1. From the Clarion IDE menu choose the **S**etup menu and then choose **T**emplate Registry.
2. When the **Template Registry** dialog opens, press the **R**egister button to open the **Template File** dialog.
3. If `C:\CW\TCTPL` is not already selected in the **D**irectories field then select it now. **TC.TPL** should now be displayed in the **F**ile **N**ame field. Select it and press the **OK** button.
4. Again select, `C:\CW\TCTPL` in the **D**irectories field. Select **TCPopup.TPL** in the **F**ile **N**ame field and press the **OK** button.
5. Notice that the new ToolCraft classes have now been added to the Template registry.
6. Press the **C**lose button to save your changes.

Complete the tutorials in the next two chapters to learn more about using the Power Browse and Power Popup products. A copy of the completed application files created by completing these exercises has been included so that you can refer to them if you have any problems. If you installed the files to the default directories, the Power Browse application file is `\CW\TCTPL\EXAMPLES\DEMO.APP` and the Power Popup application file is `\CW\TCTPL\EXAMPLES\PPDEMO`.

Working with the Power Browse Example Application

The tutorial normally included with Power Browse gives step-by-step instructions on creating the entire demo application. Since the TrialPak version of CW limits the dictionary to two files, the tutorial has been changed. You will begin by recreating the Beginning menu item procedure from the demo application. You will then complete all of the necessary additions to create the Intermediate menu item procedure.

This tutorial assumes that the Power Browse templates were installed in the directory **C:\CW\TCTPL**. If not, please be sure to substitute the appropriate drive and directory information in place of those given in the instructions.

Creating a New Dictionary

Clarion stores all of the files for a particular application in a data dictionary. To begin a new application, a dictionary containing all of the files that will be used in the application must be created. The Clarion for Windows tutorial on this CD utilizes Quick Start to create the dictionary and the application. Power Browse does include its own set of Quick Start files so that a multiple-sort order browse can be automatically created. This method will not be used for this demonstration. Instead a dictionary will be built from scratch.

The following table lists all of the necessary information needed to create the files for the example application. Follow the step-by-step instructions given to add the RECIPE file. Then use the same steps and refer to this table to add the INGREDIENT file.

Data File Name	Prefix	Field Name	Picture	Key
RECIPE	RCP	RecipeNo	N_5	Unique
		Name	S20	Unique
		Code	S5	Unique
		Type	N1	No Key
INGREDIENT	ING	RecipeNo	N_5	No Key
		Ingredient	S15	Duplicate
		Measure	S8	Duplicate
		Quantity	N2	No Key

Opening the Dictionary Editor

► To open a new dictionary

1. From the Clarion IDE menu choose the **F**ile menu and then choose **N**ew.
2. From the **New** dialog, select **Dictionary** from the **F**ile Type list.
3. Press the **OK** button to open the **Dictionary Editor**.

Now that the dictionary editor has been opened, the necessary files and their corresponding fields must be created. Files can be created in the dictionary by inserting each file and then inserting each field individually. If only basic information is needed, this method of creating a file does not need to be used. Since the example dictionary requires only very basic information, the Quick Load method will work best.

Adding Files with Quick Load

Quick Load allows the basic information about a file and its fields to be entered much more quickly than entering each piece individually.

► To access the Quick Load facility

1. Press the **Add File** button in the **Dictionary** dialog.
2. When prompted with Do you want to use Quick Load?, press the **Yes** button to open the **Clarion Quick Load** dialog.

File properties and field properties must both be defined for each of the files used in this example. Any fields in the **Clarion Quick Load** dialog which are not mentioned below should be left at the default.

► To add the file properties

1. Type *RECIPE* in the **Data File Name** field and press TAB.
2. Type *RCP* in the **Prefix** field. Press TAB twice to advance to the files list box, leaving the **File Driver** field set to **TopSpeed**.

► To add the field properties

1. Type *RecipeNo* in the **Field Name** column, then press TAB to advance to the next field.
2. Type *N_5* in the **Picture** column and press TAB.
3. Press the DOWN-ARROW to display the available choices in the **Key** column, then select **Unique** and press TAB to advance to the next line.
4. Enter information to create the Recipe Name field in the same manner as the RecipeNo field. Type *Name* in the **Field Name** column, *S20* in the **Picture** column and select *Unique* in the **Key** column.
5. To enter the Code field type *Code* in the **Field name** column, *S5* in the **Picture** column and select *Unique* in the **Key** column.
6. To enter the Type field type *Type* in the **Field name** column, *N1* in the **Picture** column and select *No Key* in the **Key** column.
7. Once the information for all four fields in the RECIPE file has been entered, press the **OK** button.
8. When prompted with Press OK to continue, or Cancel to define more fields, press the **OK** button.

Notice that the RECIPE file has been added to the **Files** list. Now go back and complete the same steps to add the INGREDIENT file using the table found at the beginning of this section. Once completed both files (RECIPE & INGREDIENT) should appear in the **Files** list.

Even though all of the basic information has been entered to create the files, the Quick Load mode is designed to enter only the essential information. More detailed information regarding the RecipeNo fields in the RECIPE & INGREDIENT files is still needed.

Adding More Information About the Fields

The end-user will not be entering the information for the recipe number field. Instead the application will take care of this (all of the steps needed to fill the RCP:RecipeNo and ING:RecipeNo fields with the correct information are given later in this exercise). For this reason, this field should utilize the read-only option. With this option, no matter where the recipe number field is used throughout the application, the end-user will not be able to change its contents.

In addition to being set to read only, the recipe number field needs to display a different prompt and column heading from those assigned to it as the default. This is more for aesthetics than for functionality. If the labels were not changed, the end user might not be clear what RecipeNo stands for. You should, therefore change it to Recipe#.

► To change information about a field

1. Select **RECIPE** from the **Files** list and press the **Fields/Keys** button.
2. Select **RCP:RecipeNo** from the **Fields** list and press the **Edit** button.
3. Type *&Recipe#:* in the **Default prompt** field.
4. Type *Recipe#* in the **Column heading** field.
5. Check the **Read only** box in the **Flags** group.
6. Press the **OK** button in the **Edit Field Properties** dialog.
7. Press the **Close** button in the **Field/Key definition** dialog.

Follow the same steps to change the ING:RecipeNo field in the INGREDIENT file.

Adding More Information About the Keys

More detailed information also needs to be added about the keys. The RECIPE file needs to have the recipe number key declared as the file's primary key. This will make the RCP:RecipeNo field the unique piece of information used to access the RECIPE file. Since the end-user will not be entering the recipe number, it also needs to be set to automatically increment when each successive record is added to the file.

► To set the Primary and Auto Number options on a key

1. Select **RECIPE** from the **Files** list and press the **Fields/Keys** button.
2. Select **RCP:Key_RecipeNo** from the **Keys** list and press the **Edit** button.
3. Check the **Auto Number** and **Primary Key** boxes in the **Attributes** group.
4. Press the **OK** button in the **Edit Key Properties** dialog.
5. Press the **Close** button in the **Field/Key definition** dialog.


The INGREDIENT file is linked to the RECIPE file by the RecipeNo field. Since the example application displays only the INGREDIENT records that are associated with the current RECIPE record being displayed, the RecipeNo field must be added to the keys for this file.

► **To create keys with multiple components (fields)**

1. Select **INGREDIENT** in the **Files** list and then press the **Fields/Keys** button.
2. Select **ING:Ingredient(ASCENDING)** listed below **ING:Key_Ingredient** in the **Keys** list, and then press the **Insert** button.
3. From the **Key Component Properties** dialog, select **ING:RecipeNo** from the list of fields. Press the **OK** button, but don't be surprised to see the dialog automatically re-open.
4. The **Key Component Properties** dialog automatically re-opens so that multiple fields can be added to the key. Since the RecipeNo field is the only one that needs to be added, press the **Cancel** button.

ING:RecipeNo(ASCENDING) has now been added to the Ingredient key. Since the INGREDIENT records need to be filtered by RecipeNo, the RecipeNo field needs to be the first component in the key (it should currently be the second).

► **To change the order of the key components**

1. Select **ING:RecipeNo(ASCENDING)** in the **Keys** list and press the up arrow  button. The RecipeNo field *must* be the first component in the key or the INSTRUCTION records cannot be displayed for each RECIPE record.
2. Follow the same procedures to add the RecipeNo field to the ING:Key_Measure key and make it the first component in the key as well.
3. Press the **Close** button to return to the list of files in the dictionary.

Adding Relationships

Since the INGREDIENT file is directly related to the RECIPE file, a relationship needs to be added to the dictionary. This relationship will establish how the child records in the INGREDIENT file will be handled when the parent record (RECIPE) is deleted. For this example, the related child records will be deleted whenever their parent record is deleted.

► **To create the RECIPE-to-INGREDIENT relationship**

1. Select **RECIPE** from the **Files** list and press the **Add Relation** button.
2. Select **Key_RecipeNo** in the **Primary Key** field.
3. Select **INGREDIENT** in the **Related File** field.
4. Select **Key_Ingredient** in the **Foreign Key** field.
5. Press the **Map By Name** button in the **New Relationship Properties** dialog.
6. Select **Cascade** in the **On Delete** field.
7. Press **OK** in the **New Relationship Properties** dialog. INGREDIENT should now be displayed in the **Related Files** list.

The dictionary for the example exercises is now complete. Using the following steps, save it before beginning the application.

Saving the Dictionary

1. Press the **Close** button to close the **Dictionary Editor**.
2. When prompted with - Do you want to save changes to UNNAMED.DCT?, press the **YES** button.
3. In the **Save Dictionary** dialog, type *DEMO.DCT* in the **File Name** field. If **C:\CW\TCTPL\TUTOR** is not already selected in the **Directories** list then select it now. Once it is selected, press the **OK** button.


The new dictionary will be saved, the Dictionary Editor will close, and you will be returned to Clarion's main menu. Continue with the exercises in the next section to create the example application which uses this new dictionary.

Creating an Application

Now that the Data Dictionary has been created, the Application Generator must be used to create the application. To begin, you need to create and name the application and tell it to use the data dictionary that you just created in the previous section.

Opening the Application Generator

► **To create a new application file**

1. From the Clarion IDE menu choose the **File** menu and then choose **New**.
2. From the **New** dialog, select **Application** in the **File Type** list and press the **OK** button.
3. When prompted with Do you want to use Quick Start, press the **NO** button. (NOTE: If you have disabled Quick Start this message will not appear. Just continue with the next step.)
4. In the **Application Properties** dialog, type *C:\CW\TCTPL\TUTOR\DEMO.APP* in the **Application File** field.
5. To enter the **Dictionary File** field, press the ellipsis  button next to the field to open the **Select Dictionary** dialog.
6. If **C:\CW\TCTPL\TUTOR** is not already selected in the **Directories** field, select it now. Once it is selected **DEMO.DCT** should appear in the **File Name** list. Select it and press the **OK** button.
7. Press the **OK** button in the **Application Properties** dialog.

Creating the MDI Frame and Menu System

The **Application Tree** should now be displayed with **Main (To Do)** selected. **To Do** is listed beside a procedure in the **Application Tree** to indicate that you need to select the procedure template to be used.

► **To create the Main procedure**

1. Press the **Properties** button in the **Application Tree** dialog. The **Select Procedure Type** dialog will list all of the available procedure templates.
2. Select **Frame-Multiple Document Main Menu** from the Clarion class and press the **Select** button.
3. When the **Procedure Properties** dialog opens press the **Window** button to open the **Window Formatter**.
4. Choose the **Menu** menu and then choose **Menu Editor**.

All of the menus and menu items are displayed in the list box on the left side of the **Menu Editor** dialog. Notice that several menus and items were automatically added. Most of these default options are not needed for the example application. Several of them need to be deleted, and some new ones added.

► **To delete menu options and items**

1. Select the menu entry labeled **&Edit** in the list box and press the **Delete** button.
2. Select the menu entry labeled **&Window** and press the **Delete** button.
3. Select the item entry labeled **P&rint Setup...** and press the **Delete** button.
4. Select the item entry labeled **SEPARATOR** and press the **Delete** button.

In addition to the default menu options remaining, the example application still needs to include the Power Browse menu option.

► **To add a new menu**

1. New menus and items are always inserted below the selected entry. Therefore, select the **END** statement right above the menu entry labeled **&Help** and press the **New Menu** button. Notice that a new menu has now been added to the list.
2. To change the default information, type *&Power Browse* in the **Text** field.
3. In the **Use** field type *?PowerBrowse*.
4. In the **Message** field type *Power Browse Example*.
5. Select the menu entry **Menu&3** in the menu list (even if it is already selected **CLICK** on it to update the information). Notice that the menu entry **&Power Browse** now appears in its place.

Now that the new menu has been added, a menu item (BrowseTest) will also be added.

► **To add a new item**

1. Select the menu entry **&Power Browse** and press the **New Item** button. Notice that a new item has now been added to the list within the Power Browse menu.
2. To change the default information, enter *&Browse Test* in the **Text** field.
3. In the **Use** field enter *?BrowseTest*.
4. In the **Message** field enter *Power Browse Test*.
5. Press the **Actions** button to open the **Prompts for ?BrowseTest** dialog.

6. Press the DOWN-ARROW to view the choices available in the **When Pressed** drop box. Highlight (select) **Call a procedure** and press ENTER. This will make the **Procedure Definition** group appear.
7. In the **Procedure Name** field type *BrowseTest*.
8. Since the application is using an MDI frame, check the **Initiate Thread** box.
9. Press the **OK** button in the **Prompts for ?BrowseTest** dialog.
10. Press the **Close** button in the **Menu Editor** dialog.
11. Press the **OK** button in the **Window Formatter**.
12. Press the **OK** button in the **Procedure Properties** dialog.

You should now be back at the **Application Tree** dialog and **Main (Frame)** should be listed instead of **Main(ToDo)**. This shows that the Main procedure has now been defined and the Frame template was used. There should also be a new menu item listed which needs to be defined.

Since the Power Browse procedure will be utilized in this application an Extension template needs to be added to define the additional global variables used in a Power Browse procedure and to allow access to the Power Browse library.

► **To add the PowerBrowseLibrary global extension template**

1. Press the **Global** button in the **Application Tree**.
2. Press the **Extensions** button in the **Global Properties** dialog.
3. Press the **Add** button in the **Edit Extensions** dialog.
4. Select **PowerBrowseLibrary - Include Power Browse Library Functions in the Application** and press the **Select** button.
5. The prompts do not need to be changed for this example, so press the **OK** button in the **Prompts for PowerBrowseLibrary** dialog.
6. Press the **OK** button in the **Edit Extensions** dialog, and then again in the **Global Properties** dialog.

Defining the Browse Procedure

The BrowseTest menu item will actually consist of two separate procedures. The first is the browse procedure which uses the Power Browse procedure template. The other is the update procedure which uses the standard Clarion form template. The update procedure allows the recipe records in the browse list box to be updated.

The **Application Tree** dialog should list the Browse Test procedure as **(To Do)**. This means this procedure still needs to be defined. The first step in defining a procedure is to select the procedure template on which it will be based. Selecting a procedure template determines what type of basic function the procedure will perform. Follow the instructions below to select the Power Browse procedure template for this procedure.

► **To select a procedure template for the browse procedure**

1. From the **Application Tree** dialog, select **BrowseTest(ToDo)**.
2. Press the **Properties** button to select the procedure template to be used for this section. The **Select Procedure Type** dialog will list all of the available procedure templates.
3. Select **Power Browse - Power Browse a Page-Loaded List Box** from the ToolCraft class and press the **Select** button.

By selecting this procedure template, several control templates were automatically added to the procedure's window properties. These controls are automatically added because they are the basic controls most often used with a browse list box. They include:

- ☐ **BrowseBox** control - The list box in which the data to be browsed will be displayed. All of the functionality of the browse list is defined on this control.
- ☐ **UpdateButtons** control - The Insert, Change and Delete buttons allow the records in the browse list box control to be updated.
- ☐ **SelectButton** control- The Select button selects a record from within the browse list box for further processing and then closes the procedure's window.
- ☐ **CloseButton** control - The Close button closes the procedure's window.

The **Procedure Properties** dialog is automatically opened when the procedure template is selected. Even though control templates were automatically added to the dialog, several additional pieces of information still need to be defined before the procedure will be complete. All of the properties related to this procedure can be accessed through the **Procedure Properties** dialog.

Not all of the properties listed in the **Procedure Properties** dialog need to be defined. Most of them allow optional features or functionality to be added to the browse procedure. For this example, only the **Files**, **Window** and **Embed** properties will be defined.

You're now ready to define the browse procedure. To start, a primary file must be defined for use with each procedure. This tells the browse list procedure which file to display information from. If only a single display order is being used then it also tells the browse list in which order the records should be displayed.

Editing the File Information

► **To select the file to be used for this procedure**

1. Press the **Files** button in the **Procedure Properties** dialog.
2. Select **<ToDo>** listed under **Browse Control** and press the **Insert** button.
3. In the **Insert File** dialog select **RECIPE** and press the **Select** button.
4. Press the **Key** button in the **File Schematic Definition** dialog.
5. Select **Key_Name** and press the **Select** button. Notice that **RECIPE - RCP:Key_Name** is now listed instead of **<ToDo>**.
6. Press the **OK** button in the **File Schematic Definition** dialog.

Now that the file information has been established, the window properties need to be defined. Each control template in the window (remember several were already added automatically) has properties and actions of its own that can be defined. The properties define the appearance of the control while the actions define the functionality. The following instructions start by explaining which properties and actions need to be defined to get the Power Browse Box functioning, and they then go on to explain each of the optional features.

To make it easier for you to get the size of the window and the browse list box set, instructions are given to enter the exact size and location of each. The window settings will be large enough to accommodate the second browse box which will be added later in this lesson. The names of the browse queue and use variable will also be changed. Since there will be multiple Browse Box controls on this window, it is easier to keep track of the lists if these names are changed. In this example they will be consecutively numbered; however for your future applications, you may choose to give them more descriptive names.

Setting the Window Sizing and Positioning

► To set the properties of the window

1. From the **Procedure Properties** dialog press the **Window** button to open the **Window Formatter**.
2. RIGHT-CLICK in the window to open the popup menu (CLICK in the window's title bar to be sure you are selecting the entire window and not an individual control) and choose **Properties**.
3. Check the **Entry Patterns** box to use the picture formats defined in the data dictionary.
4. Press the **Position** button in the **Window Properties** dialog.
5. In the **Width** group type 298 in the **Fixed** field.
6. In the **Height** group type 178 in the **Fixed** field.
7. Press the **OK** button in the **Position** dialog and again in the **Window Properties** dialog.

► To set the size and location of the browse list box

1. RIGHT-CLICK ON the Browse Box control (empty box in middle of window) and select **Position**.
2. In the **Top Left Corner** group, type 4 in the **X** field and 29 in the **Y** field.
3. In the **Width** group type 149 in the **Fixed** field.
4. In the **Height** group type 73 in the **Fixed** field.
5. Press the **OK** button in the **Position** dialog.

The Close Button control needs to be moved to make room for the second list box which will be added later. The Select Button which was automatically populated with the browse control will not be used in this exercise and can be deleted.

► To move a control

1. CLICK-DOWN on the Close button control (the cross cursor will appear).
2. Drag the control to the top right corner of the Window and RELEASE. (**NOTE:** You may have to move the **Controls** toolbox out of the way to see what you are doing)

► **To delete a control**

1. Since the **Select** button will not be used it should be deleted (this will remove all of the code associated with it from this procedure). **CLICK** on the **Select** button and press **DELETE**.
2. When prompted with Deleting this will delete the control template, press the **OK** button.
3. When prompted with Do you want to delete any associated controls?, press the **YES** button.

Setting the Browse Box Control's Properties

The browse procedure's browse list box has several navigational controls. The horizontal and vertical scroll bars allow a mouse to be used to move the display in the browse box right and left or up and down. The VCR controls function in the same manner as video cassette recorder controls and allow a mouse to be used to move to the top or bottom of the list, to move the display up or down a page, or to move up or down one entry. Follow the directions below to add all three of these options.

► **To set the properties for the browse list box**

1. **RIGHT-CLICK** on the Browse Box control and choose **Properties**.
2. Click the **Horizontal**, **Vertical** and **VCR** boxes to add all of the navigational controls.

Although the browse procedure already knows to use the **RECIPE** file as its source of information (this was established when the **Files** properties were set), it still does not know which pieces of data to display or in what format. To define the format of the browse list box control, fields must be added (populated) from the **RECIPE** file to the browse list box.

► **To populate the browse list box with data**


1. Press the **Format** button in the **List Properties** dialog.
2. Press the **Populate** button in the **List Box Formatter** dialog.
3. In the **Select Field** dialog, **RECIPE - RCP:Key_Name** should already be selected in the **Files** list. If it is not, select it now.
4. Select **RCP:Name** from the **Fields** list and press the **Select** button.
5. Type *I* in both **Indent** fields.
6. Check the **Underline** and **Right Border** boxes.
7. Press the **OK** button in the **List Field Properties** dialog.
8. Press the **Populate** button again to add the next field.
9. Select **RCP:Code** from the **Fields** list and press the **Select** button.
10. You need to make this field wider by typing *24* in the **Width** field.
11. Notice that the **Underline**, **Right Border** and the **Indent** fields are already defaulted to what you entered for the last field. Press the **OK** button in the **List Field Properties** dialog.
12. Press the **Populate** button one more time to add the last field.
13. Select **RCP:Type** from the **Fields** list and press the **Select** button.
14. Since no changes are needed, just press the **OK** button in the **List Field Properties** dialog.

15. Press the **OK** button in the **List Box Formatter** dialog.
16. Press the **OK** button in the **List Properties** dialog.
17. Press the **OK** button to close the **Window Formatter**.
18. Press **OK** in the **Procedure Properties** dialog.

Notice that (ToDo) is no longer listed beside BrowseTest. Instead, PowerBrowse is now listed to show that the Power Browse procedure template was selected for use in this procedure.

Enough information has now been entered about the browse procedure to go ahead and test it. Before you do this, the application must first be generated.

► **To generate the application**

1. CLICK the Make  button on the Clarion IDE toolbar.
2. Information letting you know that the application is being generated will be displayed. Once the application has been generated, a green check mark will appear in the bottom right corner of the window. If a red X appears, you made a mistake completing the exercise. Go back through the exercise to locate the problem or try looking at the on-line help **Troubleshooting** section to locate the problem.
3. Press the **Run** button to open the application's main menu.

Throughout the lessons you will be instructed to test the application. It will be much easier to locate a problem if you check your work periodically, rather than waiting until you are completely finished with the exercise. Follow the instructions below to view your progress so far.

► **To view your progress on the browse procedure**

1. Choose the **Power Browse** menu and then choose **Browse Test**.
2. Press the OK button when the TrialPak warning message is displayed.
3. Notice that the **Type** field in the browse list box displays 1, 2 or 3. You should recall that the example application displayed Main, Side or Dessert in this field. The RCP:Type field was declared with a format of N1, not S8 (which would store the actual type) and contains a number which represents the recipe type instead of the actual type itself. Instructions will be given later to show how to display the appropriate information.
4. Test all of the features which you just added, such as the navigational controls, and make sure that everything appears to be working correctly.
5. When you are finished looking over your progress, choose the **File** menu and then choose **Exit**.

Now that you have the basic features of the browse list box working, you are ready to get back into the application and begin adding some optional features.



► **To edit the browse procedure**

1. Select **BrowseBeginning(PowerBrowse)** and press the **Properties** button.
2. Press the **Window** button in the **Procedure Properties** dialog.

Adding a Search Field

A search entry field needs to be added to the window to search for specific entries in the browse list box.

► To add a search entry control template to the screen

1. Select the Control Template  tool from the **Controls** toolbox.
2. Select **SearchEntryAndPrompt - Search Entry Field and Prompt** from the ToolCraft class and press the **Select** button.
3. Move the populate  cursor to the location where the search field needs to be placed (just above the left corner of the Browse Box control) and **CLICK** to place it there.
4. Both the search prompt and entry field should now be placed. Rearrange them if necessary by selecting each one and dragging it to the desired location.

Now that the Search Entry control has been added to the window properties, the Browse Box needs to know that it should allow entry searches and step searches. The name of the search entry field to be used also needs to be entered.


► To set the Power Browse actions to use both locator types

1. **RIGHT-CLICK** the Browse Box control and choose **Actions**.
2. In the **Locator** field select **Both** from the list. This will allow both a step search and an entry search to be processed for the same browse list box.
3. Select **?Search** in the **Search Input Field** field.
4. Press the **OK** button in the **Prompts for ?List** dialog.

Adding Hot Display Fields

Hot display fields are display-only fields which are automatically re-displayed each time a new entry is selected in the browse list box. These fields are populated from the file being displayed in the browse list box.

► To add the fields to hot-display to the screen

1. Select the Multi-Populate  tool from the **Controls** toolbox. The Select Field dialog will open. Each time you select a field, it will automatically re-open to allow you to choose additional fields.
2. Select **RECIPE - RCP:Key_Name** in the **Files** list.
3. Select **RCP:RecipeNo** in the **Fields** list and press the **Select** button.
4. Use the populate cursor to position the field in the blank space below the Browse Box control and **CLICK** to place it there. Leave room just below the update buttons so that the mark buttons can be added later.
5. When the **Select Field** dialog re-opens, select **RCP:Name** and press the **Select** button.
6. Place it on the window just below the recipe number hot display field in the same manner.
7. Once these two fields have been placed on the window, press the **Cancel** button in the **Select Field** dialog to quit picking fields.
8. Rearrange the fields that were just placed if necessary.

Since these fields are for displaying information only, they need to have the read-only attribute assigned. You should recall that this attribute was already added to the RCP:RecipeNo field when it was originally added to the Data Dictionary; however, the RCP:Name field does not have this attribute assigned.

► **To make hot display fields read-only**

1. RIGHT-CLICK on the RCP:Name entry field (not the prompt) and choose **P**roperties.
2. Check the **Read Only** box and press the **O**K button.

Now that the fields have been added to the screen, the browse list box needs to know to display them as hot fields.

► **To set the Power Browse actions to display hot fields**

1. RIGHT-CLICK the Browse Box control and choose **A**ctions.
2. Check the **Enable "Hot" Display Fields** box and press the **O**K button.

Adding a Local Variable to Display in the List

As pointed out when you tested the browse procedure, the RCP:Type field contains a number (1,2,3) to represent Main, Side or Dessert. Several steps need to be taken to make the browse list display the appropriate type. The first is to delete RCP:Type from the list of fields to display in the browse list box. It was added to the browse list box just to demonstrate that it would not display the desired information.

► **To delete a display field from the list box**

1. RIGHT-CLICK the Browse Box control and select **P**roperties.
2. Press the **F**ormat button to open the **List Box Formatter** dialog.
3. Select Field Number 3 (**RCP:Type**) by pressing the RIGHT-ARROW twice. The column titled **T**ype should now be selected.
4. Press the **D**efine button.

Next, create a variable to use for displaying the information in the browse list box. A local variable will be added for this purpose. Since this variable will also need to be added to the browse list box to replace the RCP:Type field that was just deleted, it can be added directly from the **List Box Formatter**.

► **To add a local variable and include it in the list box**

1. Press the **P**opulate button in the **List Box Formatter** dialog.
2. Select **LOCAL DATA BrowseTest** from the **F**iles list.
3. Press the **N**ew button to add a new local variable.
4. Type *TypeDsp* in the **N**ame field.
5. Type 7 in the **C**hars field.
6. Type *&Type:* in the **D**efault prompt field.
7. Type *Type* in the **C**olumn heading field.

8. Press the **OK** button in the **New Field Properties** dialog and again in the **List Field Properties** dialog.
9. Press the **OK** button in the **List Box Formatter** dialog.
10. Press the **OK** button in the **List Properties** dialog.
11. Press the **OK** to close the **Window Formatter**.

Although a string variable has now been placed in the List Box Formatter, the browse list box still doesn't know what to display in the field. If you were to run the application now, the **Type** column in the browse list box would be blank. The final step needed to display the recipe type is to add an embed point.

Using an Embed Point to Format an Element

A few lines of code need to be added to tell the browse list how to format the recipe type. The templates are only coded to handle basic situations; therefore it is sometimes necessary to add additional code to account for special situations. This is why embed points are set up which allow code to be easily inserted in pre-defined locations. An embed point is available for **Formatting an element in the browse queue**, as is needed in this situation.

► To add an embed point for formatting an element

1. Press the **Embeds** button in the **Procedure Properties** dialog.
2. In the **Embedded Source** dialog look for **Format an element of the browse queue** about half way down the list of embed points and select **Browse List Box(1)**, which should be directly below it.
3. Press the **Add** button in the **Embedded Source** dialog.
4. Select **SOURCE** and press the **Select** button.
5. The text editor will open and display a blank screen. The code needed to display the correct recipe type needs to be added here. Enter the following:


```
CASE RCP:Type
  OF 1
    TypeDsp = 'Main'
  OF 2
    TypeDsp = 'Side'
  OF 3
    TypeDsp = 'Dessert'
END
```
6. When you are done entering the code, choose the **File** menu from the Editor's Main menu and then choose **Close**.
7. When prompted with **Save "Format an element of the browse queue (Browse List Box(1))"**, press the **Yes** button.
8. Press the **Close** button in the **Embedded Source** dialog.

Now that everything in the browse list box will display as it should and all of the added features are set up, the last thing to define for this browse list box is how the records in the list will be updated.

Defining the Update Process

An update process needs to be established for updating the RECIPE records that are listed in the browse list box. Two options are available. The first is to create a form update procedure which will open each time a RECIPE record is inserted, changed or deleted. It will contain entry fields for entering all of the information included in the RECIPE file. The other option is in-line editing. This option allows all of the fields in the browse list box to be edited directly in the browse list box. For this example, the form update method will be used.

► To add a form update procedure to a Browse Box

1. Press the **Window** button in the **Procedure Properties** dialog.
2. RIGHT-CLICK the Browse Box control and choose **A**ctions.
3. Make sure that the **Use In-Line Data Entry** box is *not* checked.
4. In the **Update Procedure** field type *UpdateRecipe*.
5. Press the **OK** button in the **Prompts for ?List** dialog.
6. Press the **OK** button to close the **Window Formatter**.
7. Press the **OK** button in the **Procedure Properties** dialog.

This procedure will be added to the Application Tree with **(To Do)** listed beside it. It will be defined in the following section. Before moving on to the next section, you should test your work again.

► To test your work

1. Press the Make button in the Clarion IDE toolbar.
2. Remember, once the green check mark appears, press the **R**un button to start the application.
3. Test the newly added features to make sure they are working then exit the example application.

If everything appears to be working correctly, move on to the next section to define the UpdateRecipe procedure.

Defining the Update Recipe Procedure

Since a form was defined for use in the Browse Box's actions, a new procedure was created. **UpdateRecipe (To Do)** should now appear in the **Application Tree** dialog and needs to be defined. Since the procedure uses the standard form template, only the essential information to create this screen will be given here. If you want more information, refer to the Clarion manuals.

A procedure template must be selected as the basis for the procedure, just as for the browse procedure. The form template, like the power browse template, places several control templates in the procedure's window properties automatically. They include an OK and Cancel button and a message field.

- **To select the template for the update recipe procedure**
 1. In the **Application Tree** dialog, select **UpdateRecipe(ToDo)**.
 2. Press the **Properties** button to select the procedure template to be used for this section.
 3. Select **Form - View/Edit a Record from a File** from the Clarion class and press the **Select** button. The **Procedure Properties** dialog will automatically open.

Just as the browse procedure required a file to be designated, so does the form procedure. This will tell the form which file will be updated.

Editing the File Information

- **To select the file to be used for this procedure**
 1. Press the **Files** button in the **Procedure Properties** dialog.
 2. Select <ToDo> listed under **Update Record on Disk** and press the **Insert** button.
 3. In the **Insert file** dialog select **RECIPE** and press the **Select** button.
 4. Press the **OK** button in the **File Schematic Definition** dialog.

You need to make sure that all of the information to be entered on this form will be entered in the correct format by the end-user. When you created the dictionary, each field had an entry picture specified for use on the window. If you don't specify that these entry picture formats should be used, they will be ignored.

- **To use the entry patterns defined for each field**
 1. Press the **Window** button in the **Procedure Properties** dialog.
 2. RIGHT-CLICK the window's title bar and choose **Properties**.
 3. Type *Update Recipe Records* in the **Title** field.
 4. Check the **Entry Patterns** box.
 5. Press the **Position** button in the **Window Properties** dialog.
 6. Press the **OK** button in the **Window Properties** dialog.



Unlike those in the browse procedure, none of the control templates automatically placed on the form window needs to have either its properties nor actions changed. The window, however, does need to be populated with the fields to be entered for the RECIPE file.

- **To populate the update form with entry fields**
 1. Select the Multi-Populate tool from the **Controls** toolbox to open the **Select Field** dialog.
 2. **RECIPE - RCP:Key_Name** should be selected in the **Files** list. If it is not, select it now.
 3. Select **RCP:RecipeNo** from **Fields** list and press the **Select** button.
 4. Use the populate cursor to place the entry field at the top of the window.

5. The multi-populate mode will automatically return to the **Select Field** dialog each time you place a field. Continue in the same manner and place the RCP:Name and RCP:Code fields.
6. When the three entry fields have been placed, press the **Cancel** button in **Select Field** dialog to exit the multi-populate mode.

The RCP:Type field was not populated directly from the RECIPE file to the window because it contains a number instead of the actual type. Instead of typing in the recipe type, a list box will be used to select Main, Side or Dessert.

► **To add a list box to select the recipe type**

1. Select the Prompt  tool from the **Controls** toolbox.
2. Use the cross-hairs cursor to place the prompt below the other fields just placed. While the prompt control is still selected, type *&Type:* in the **Caption** field which is located in the toolbar at the top of the window.
3. Type *?TypeList:Prompt* in the **Use** field.
4. Select the List Box  tool from the **Controls** toolbox.
5. Use the cross-hairs cursor to place the list box next to the prompt control.
6. The **List Box Formatter** will automatically open. No changes need to be made, so just press the **OK** button.
7. RIGHT-CLICK ON the new list box and choose **Properties**.
8. Type **Main|Side|Dessert** in the **From** field. The pipe (|) symbol is used to separate each choice in the list box.
9. Type *?TypeList* in the **Use** field.
10. Type 3 in the **Drop** field.
11. This list box needs to be the same size as the other entry fields. Press the **Position** button.
12. Type 12 in the **Fixed** field located in the **Height** group.
13. Press the **OK** button in the **Position** dialog and again in the **List Properties** dialog.
14. Press the **OK** button to close the **Window Formatter**.

Since the recipe type list box was not populated directly from the RECIPE file, whatever choice the user selects in this field will not automatically be saved to the RECIPE file. An embed point will need to be added to save the number representing the recipe type into the RCP:Type field.

► **To add an embed point to save the RCP:RecipeNo field**

1. Press the **Embeds** button in the **Procedure Properties** dialog.
2. Look for **Control Event Handling, After Generated Code** in the list of embed points and CLICK the plus sign (+) beside *?TypeList* to view all of the actions for this field.
3. Select **Accepted** and press the **Add** button.
4. Select **SOURCE** and press the **Select** button.

5. The code needed to save the number representing the recipe type selected needs to be added here. Type the following: *RCP:Type = CHOICE()*
6. Choose the **F**ile menu and then choose **C**lose.
7. When prompted with Save "Control Event Handling, After Generated Code(?Typelist,Accepted)", press the **Y**es button.

Another embed point also needs to be added to set the **?TypeList** list box to the appropriate selection each time the update form is opened. If this code were not entered then the **Type** field would always be blank when the form was initially opened.


► **To add an embed point to select an entry in a list box**

1. On the **Embedded Source** dialog select **Preparing to Process the Window** in the list of embed points.
2. Press the **A**dd button in the **Embedded Source** dialog.
3. Select **SOURCE** and press the **S**elect button.
4. Type the following code in the editor window:


```
IF RCP:Type
    SELECT(?TypeList,RCP:Type)
ELSE
    SELECT(?TypeList,1)
    RCP:Type = 1
END
```
5. Choose the **F**ile menu and then choose **C**lose.
6. When prompted with Save "Preparing to Process the Window", press the **Y**es button.
7. Press the **C**lose button in the **Embedded Source** dialog.
8. Press the **O**K button in the **Procedure Properties** dialog.

UpdateRecipe(Form) should now appear in the **Application Tree** list instead of **(ToDo)**. Save and test your work again.

► **To save and test your work**

1. Press the Save  button in the Clarion IDE toolbar.
2. Press the Make button in the Clarion IDE toolbar.
3. Remember, once the green check mark appears, press the **R**un button to start the application.
4. Test all of the features to make sure everything is working the way it is supposed to. Once you are done exit the example application.

You are now ready to add the features found in the Intermediate section of the demo application. You have already saved your work. If you prefer, you can take a break now and complete the remaining exercises later.

Adding the Intermediate Browse Features

The browse box that was just created in the previous exercises will be changed to allow multiple sort orders and two of these sort orders will utilize the filter feature to display only a specified set of records.

Adding Multiple Sort Orders and Using Ranges

► **To add the Multiple Sort Order Control**

1. Select **BrowseTest(PowerBrowse)** and press the **Properties** button.
2. Press the **Window** button in the **Procedure Properties** dialog.
3. Select the Control Template tool from the **Controls** toolbox.
4. Select **OrderDropBox - Change Order Drop Box** from the ToolCraft Class and press the **Select** button.
5. Place the drop box just above the Search Entry control.
6. The **List Box Properties** dialog will automatically open. Nothing needs to be changed, so just press the **OK** button.

The multiple display orders feature requires more information to be entered in the Browse Box control's **Actions** than any of the previously discussed controls. The display format for a browse list box is usually entered in the **List Box Formatter** dialog accessed through the **Format** button in the Browse Box control's **Properties**. When multiple display orders are used, that format is ignored and a display format must be defined for each individual sort order from within the Browse Box control's **Actions**.

► **To set the browse list box actions to use multiple sort orders**

1. RIGHT-CLICK on the Browse Box control and choose **Actions**.
2. Check the **Use Multiple Display Orders** box.
3. When the **Display Orders** button appears, press it.

Each display order must be added and then defined in the **Display Orders** dialog. There are actually two dialog windows titled **Display Orders**. The first one is where the actual list of display orders can be inserted, changed or deleted. This dialog is referred to as the **Display Orders** dialog. The second is where all of the details for each display order are defined. This dialog is referred to as the **Display Orders Format** dialog.

► **To add new display orders**

1. In the **Display Orders** dialog, you can insert as many display orders as you want. Press the **Insert** button to open the **Display Orders Format** dialog.
2. Type *By Name* in the **Order Description** field.
3. To select the key to be used with this sort order, press the ellipsis button next to the **Key Name** field.
4. Select **RECIPE - RCP:Key_RecipeName** in the **Files** list and **RCP:Key_Name** in the **Keys** list.
5. Press the **Select** button in the **Select Key** dialog.

This list box was already populated with the fields to be displayed from the RECIPE file. However, for multiple sort orders, that display order is ignored and the same type of process needs to be completed for each individual sort order.

► **To populate the multiple sort order List Box Formatter**

1. Press the ellipsis button next to the **List Box Format** field.
2. The populate button is missing from this **List Box Formatter** dialog because it is currently not a function available to third-party templates. Instead, each field will have to be inserted. Press the **Insert** button.
3. The **List Field Properties** dialog will open. Normally if you had used the populate key all of this information would have automatically been entered. When using multiple sort orders you need to enter this information yourself. Type 80 in the **Width** field.
4. Type @S20 in the **Picture** field.
5. Type *Name{RCP:Name}* in the **Heading text** field. Since the **List Field Properties** dialog does not provide a place to enter the field to display, both the heading and the field name must be entered in the **Heading text** field. As shown in this entry, the heading is entered first and then the field name is entered in braces immediately after.
6. Type 1 in both **Indent** fields.
7. Check the **Underline** and **Right Border** boxes.
8. Press the **OK** button in the **List Field Properties** dialog.
9. Press the **Insert** button again to add another field.
10. Type 20 in the **Width** field.
11. Type @S5 in the **Picture** field.
12. Type *Code{RCP:Code}* in the **Heading text** field.
13. Notice that, just as when you used the populate button in the beginning exercise, the **Underline** and **Right Border** boxes are already checked, and the **Indent** fields are already set. Just press the **OK** button in the **List Field Properties** dialog.
14. Press the **OK** button in the **List Box Formatter** dialog.
15. Type @S20 in the **Search Field Picture** field. This lets the Search Entry control know what picture format to use. For example, you may need to sort by date (@D1) in one sort order and a string (@S5) in another. This is a required field when the search **Locator** type is set to **Entry**, **Both** or **Incremental**. If it is set to **Step** or **None** then this field is disabled.
16. Press the **OK** button in the **Display Orders** Format dialog.

Now that you are back on the **Display Orders** dialog, you need to go back and repeat this process to add the Beef Dishes Only and then the Chicken Dishes Only sort orders. This time, The @USE#@ feature will be used which will cause these two sort orders to use the same list box format string as the first sort order. This means that you will not have to insert all of the fields again in the list box formatter.

1. Press the **Insert** button to open the **Display Orders** dialog.
2. Type *Beef Dishes Only* in the **Order Description** field.

3. To select the key to be used with this sort order, press the ellipsis button next to the **Key Name** field.
4. Select **RECIPE - RCP:Key_RecipeName** in the **Files** list and **RCP:Key_Name** in the **Keys** list.
5. Press the **Select** button in the **Select Key** dialog.
6. Press the ellipsis button next to the **List Box Format** field and then press the **Insert** button.
7. Type `@USE#1@` in the **Heading text** field.
8. Press the **OK** button in the **List Field Properties** dialog.
9. Type `@S20` in the **Search Field Picture** field.
10. Press the **OK** button in the **Display Orders** Format dialog.

Repeat all of the steps above to add the Chicken Dishes Only display order, just type *Chicken Dishes Only* in the Order Description field instead of Beef Dishes Only.

Once you have finished adding all three sort orders, they should all appear in the **Display Orders** dialog. If you left the sort orders as is, all three would display the exact same information in the exact same way. Since the last two sort orders need to display different information, you need to go back in and add the information needed to use record filtering. These two display orders should display only those records with either Beef or Chicken in the RCP:Name field.

► **To use record filtering on a sort order**

1. Select **Beef Dishes Only** from the **Display Orders** dialog and press the **Edit** button.
2. Type `INSTRING('BEEF',UPPER(RCP:Name),1)` in the **Record Filter** field. Make sure you type this exactly as shown, paying particular attention to upper and lower case letters.
3. Press the **OK** button on the **Display Orders** Format dialog.
4. Select **Chicken Dishes Only** from the **Display Orders** dialog and press the **Edit** button.
5. Type `INSTRING('CHICKEN',UPPER(RCP:Name),1)` in the **Record Filter** field.
6. Press the **OK** button in the **Display Orders** Format dialog.
7. Press the **OK** on the **Display Orders** dialog.

The last piece of information that the browse list box needs to know is the name of the sort order control that will be used to select sort orders.

► **To select the sort order control for this browse list box**

1. Select **?ChangeOrder** in the **Change Order Control** field.
2. Press the **OK** button in the **Prompts for ?List** dialog.

Adding Mark Buttons

Mark buttons will be added in this example just to show how they work. Once the records are marked they will not be processed, however you could process them in whatever manner you choose (i.e. delete them, print them, etc.)

The MarkButtons control consists of the **Mark**, **Unmark**, **Mark All**, and **Clear All** buttons.

NOTE: To use the Mark records control template, the file being displayed in the browse list box must have a primary key defined. Since the RECIPE file already has one you don't need to do anything now, but keep this in mind when developing your own applications.

► **To add the MarkButtons control**

1. Select the Control Template tool from the **Controls** toolbox.
2. Select **MarkButtons - Record Marking Buttons** from the ToolCraft Class and press the **Select** button.
3. Place the Mark Buttons just below the Update Buttons.

Notice that there is not enough room to fit all four Mark Buttons evenly below the bottom of the Browse Box control. The default width of the buttons is set to 40. Complete the following steps for each of the four Mark Buttons to change the width to 35.

► **To change the size of the Mark Buttons**

1. RIGHT-CLICK on the Mark Button control and choose **Position**.
2. Type 35 in the **Fixed** field in the **Width** group.
3. Press the **OK** button in the **Position** dialog.
4. Repeat these steps for the other three Mark Button controls.
5. Once all four buttons are changed line them up evenly under the Browse Box control.


The **Actions** for all four Mark Buttons are the same. If you change the actions on any one of the four buttons it changes the actions for all of them. In this example you will be changing the actual mark string which will be displayed in the browse list box.

► **To change the mark string**

1. RIGHT-CLICK on the Mark Button control and select **Actions**.
2. Press the ellipsis button next to the **List Box Display Field** field.
3. Select **LOCAL DATA BrowseTest** in the **Files** list.
4. Select **MarkStatus** in the **Fields** list and press the **Select** button. This variable was automatically added when the Mark Buttons control template was selected.
5. Type *Marked* in the **Marked Message** field.
6. Type *UnMarked* in the **Not Marked Message** field.
7. Press the **OK** button on the **Prompts for ?Mark** dialog.

The only step needed to let the browse list box know to use the Mark Buttons is to actually populate the **List Box Formatter** with the local data variable named in the **List Box Display Field** field in the **Prompts for ?Mark** dialog. This variable was automatically created when you picked the Mark Buttons control. It will display the mark string (Marked or UnMarked) in the browse list box. Since the second and third sort orders are using the same format as the first, you need only to add this field to the first display format.

► **To add the MarkStatus field to each display order format**

1. RIGHT-CLICK on the Browse Box control and choose **A**ctions.
2. Press the **Display Orders** button.
3. Select **B**y Name and press the **E**dit button.
4. Press the ellipsis button next to the **List Box Format** field.
5. Press the **I**nsert button in the **List Box Formatter** dialog.
6. Type *34* in the **Width** field.
7. Type *@S8* in the **P**icture field.
8. Type *Mark{MarkStatus}* in the **H**eading text field.
9. Press the **O**K button in the **List Field Properties** dialog
10. Use the move left  button to move the Mark column to the first position in the list.
11. Press **O**K in the **List Box Formatter** dialog.
12. Press **O**K in the **Display Orders** Format dialog and again in the **Display Orders** dialog.
13. Press **O**K in the **P**rompts for ?List dialog.
14. Press **O**K to close the **Window Formatter**.
15. Press **O**K in the **Procedure Properties** dialog.

The first browse box is now complete, so this would be a good time to save and test your work again.

Now that the first browse list box is working, you need to go back in and add the second browse list box and its related controls. When more than one browse list box control is used in the same window, some of the related controls require additional information to be entered. Please be sure to read through these instructions carefully so that you don't overlook these additional steps.

► **To get back into the BrowseTest procedure**

1. Select **BrowseTest(PowerBrowse)** and press the **P**roperties button.
2. Press the **W**indow button in the **Procedure Properties** dialog.

Adding a Second Browse Box Control

The second browse list box in this window will display the INGREDIENT records that are related RECIPE record currently selected in the first browse list box.

► **To add the Browse Box control**

1. Select the Control Template tool from the **C**ontrols toolbox.
2. Select **BrowseBox - Browse List Box** from the ToolCraft Class and press the **S**elect button.
3. Place the second browse list box beside the first one.
4. When the **List Box Formatter** dialog opens, press the **O**K button.

► **To set the properties for the browse list box**

1. RIGHT-CLICK on the second Browse Box control and choose **Properties**.
2. Type *?List2* in the **Use** field.
3. Check the **Hide Selection** and **Vertical** boxes.
4. Press the **Position** button in the **List Properties** dialog.
5. In the **Top Left Corner** group, type *160* in the **X** field and *29* in the **Y** field.
6. In the **Width** group type *133* in the **Fixed** field.
7. In the **Height** group type *73* in the **Fixed** field.
8. Press the **OK** button in the **Position** dialog.
9. Press the **OK** button in the **List Properties** dialog.

When you added the second Browse Box control, only the box itself was placed. Each of the related controls will need to be added to the window individually. The same set of update buttons could be used for both this list box and the one already created, however for this example each list box will use its own set.

Adding Update Buttons to the Second List Box

► **To add the Update Buttons Control**

1. Select the Control Template tool from the **Controls** toolbox.
2. Select **UpdateButtons - Update Records from a Browse Box Control** from the ToolCraft Class and press the **Select** button.
3. Something a little different will happen here. The **Select Parent** dialog will open. This dialog opens whenever another set of Update Buttons has already been placed on the window. Choose **Browse List Box(7)** and press the **Select** button. (NOTE: To keep track of the Browse Box control's numbers, keep in mind that they are numbered in the order in which they were added to the window. The number is Clarion's control template instance number.)
4. Place the buttons directly below the second Browse Box control.

Since you already established which Browse Box control these Update Buttons are associated with when you selected the Browse Box control from the **Select Parent** dialog, nothing else needs to be done.

Adding Hot Display Fields to the Second List Box

The hot display fields do not require any additional steps when used with more than one browse list box. The fields are just automatically updated whenever the file from which they are populated displays a new record in a browse list box. Just as in the other examples, all you need to do is populate the fields to the window, make them read-only and then check the hot display box in the Browse Box control's **Actions**.


► **To add hot display fields**

1. Select the Multi-field Populate tool from the **Controls** toolbox.
2. Notice that **<ToDo>** is listed for the second Browse Box control file. The file can be selected here rather than from the **Files** properties. You take the same steps that you would have taken if you had pressed the **Files** button from the **Procedure Properties** dialog.

3. Select <**ToDo**> and then press the **I**nsert button.
4. Select **INGREDIENT** from the Files list and then press the **S**elect button.
5. Select **ING:RecipeNo** in the **F**ields list and press the **S**elect button.
6. Place the field in the space below the second Browse Box control.
7. Select **ING:Ingredient** and place it on the screen in the same manner.
8. Press the **C**ancel button in the **S**elect **F**ield dialog to quit picking fields.
9. RIGHT-CLICK on the ING:Ingredient entry field and choose **P**roperties.
10. Check the **R**ead **O**nly box and press the **O**K button.
11. Go to the actions for the second browse list box (remember, RIGHT-CLICK and choose **A**ctions) and check the **E**nable "**H**ot" **D**isplay **F**ields box.
12. Press the **O**K button in the Prompts for **?List2** dialog.

Adding Multiple Sort Orders to the Second List Box

► To add the Order Option Box control

1. Select the Control Template tool from the Controls toolbox.
2. Select **OrderOptBox - Change Order Option Box** from the ToolCraft Class and press the **S**elect button.
3. Place the option box above the second Browse Box control.
4. RIGHT-CLICK the Sort By option box and choose **P**roperties.
5. Delete the text in the **C**aption **T**ext field.
6. Type *?ChangeOrder2* in the **U**se field.
7. Uncheck the **B**oxed box.
8. Press the **O**K button in the **O**ption **P**roperties dialog.
9. Select the Radio Button  tool from the **C**ontrols toolbox.
10. Place the button in the Order Option Box control.
11. RIGHT-CLICK the **R**adio**1** radio button and choose **P**roperties.
12. Type *Ingredient* in the **B**utton **T**ext field.
13. Press the **O**K button in the **R**adio **B**utton **P**roperties dialog.
14. Select the Radio Button tool again from the **C**ontrols toolbox.
15. Place the button in the Order Option Box control beside the other radio button.
16. RIGHT-CLICK the **R**adio**2** radio button and choose **P**roperties.
17. Type *Measure* in the **B**utton **T**ext field.
18. Press the **O**K button in the **R**adio **B**utton **P**roperties dialog.

The browse list box **A**ctions for this control are defined in the exact same manner as those for the other sort order controls. Therefore, if you change your mind and decide to use another type of control, all you have to do is replace this control and select the new one in the **Change Order Control** field.

► **To populate the list box with sort orders**

1. RIGHT-CLICK the second Browse Box control and choose **A**ctions.
2. Check the **Use Multiple Display Orders** box.
3. Press the **Display Orders** button.
4. Press the **I**nsert button in the **Display Orders** dialog.
5. Type *By Ingredient* in the **Order Description** field.
6. Press the ellipsis button next to the **K**ey Name field.
7. Select **INGREDIENT - ING:Key_Inгредиент** in the **Files** list and **ING:Key_Inгредиент** in the **Keys** list.
8. Press the **S**elect button in the **Select Key** dialog.
9. Press the ellipsis button next to the **List Box Format** field.
10. Press the **I**nsert button in the **List box Formatter** dialog.
11. Type *60* in the **Width** field.
12. Type *@S15* in the **Picture** field.
13. Type *Ingredient{ING:Ingredient}* in the **Heading text** field.
14. Type *1* in both **Indent** fields and check the **U**nderline and **R**ight **B**order boxes.
15. Press the **O**K button in the **List Field Properties** dialog.
16. Press the **I**nsert button again to add another field.
17. Type *32* in the **Width** field.
18. Type *@S8* in the **Picture** field.
19. Type *Measure{ING:Measure}* in the **Heading text** field.
20. Press the **O**K button in the **List Field Properties** dialog.
21. Press the **I**nsert button one last time.
22. Type *20* in the **Width** field.
23. Type *@S5* in the **Picture** field.
24. Type *Qty{ING:Quantity}* in the **Heading text** field.
25. Press the **O**K button in the **List Field Properties** dialog.
26. Press the **O**K button in the **List Box Formatter** dialog.
27. Press the **O**K button in the **Display Orders** Format dialog.

28. Repeat the same process to add the **By Measure** sort order. Replace *By Ingredient* with *By Measure* when entering the **Order Description** field. Select **ING:Key_Measure** in the **Key Name** field. When inserting the fields in the **List Box Formatter** dialog, change the order to Measure, Quantity and then Ingredient.
29. Press the **OK** button in the **Display Orders** dialog.
30. Select **?ChangeOrder2** in the **Change Order Control** field to tell the browse to use the newly added control to change sort orders.

This browse list box will be updated using the in-line editing feature instead of a form. Although multiple display orders will be used with this browse list box, each field needs to be entered in the in-line entry field information only once.

Adding In-Line Entry Updating

► To set the Browse Box to use in-line entry

1. Check the **Use In-Line Data Entry** box.
2. Press the **In-Line Fields** button.
3. Press the **Insert** button in the **In-Line Fields** dialog.
4. Press the ellipsis button next to the **Field Name** field.
5. Select **INGREDIENT - ING:Key_Inгредиент** in the **Files** list.
6. Select **ING:Ingredient** in the **Fields** list and press the **Select** button.
7. Select **Entry** in the **Field Type** field.
8. Press the **OK** button in the **In-Line Fields Format** dialog.
9. Press the **Insert** button in the **In-Line Fields** dialog.
10. Press the ellipsis button next to the **Field Name** field.
11. Select **INGREDIENT - ING:Key_Inгредиент** in the **Files** list and **ING:Measure** in the **Fields** list.
12. Press the **Select** button in the **Select Field** dialog.
13. Select **Drop List** in the **Field Type** field.
14. Type 7 in the **Drop** field.
15. Type *'cup/gallon/lb/ounce/pint/tbl/tsp'* in the **From** field.
16. Type *32L@S8@* in the **Format** field (you could press the ellipsis button to use the List Box Formatter if you prefer).
17. Press the **OK** button in the **In-Line Fields Format** dialog.
18. Press the **Insert** button in the **In-Line Fields** dialog.
19. Press the ellipsis button next to the **Field Name** field.
20. Select **INGREDIENT - ING:Key_Inгредиент** in the **Files** list and **ING:Quantity** in the **Fields** list.

21. Press the **Select** button in the **Select Field** dialog.
22. Select **Spin Box** in the **Field Type** field.
23. Type *I* in the **Initial Value** field.
24. Type *99* in the **Upper** field.
25. Type *I* in the **Lower** field.
26. Type *I* in the **Step** field.
27. Press the **OK** button in the **In-Line Fields Format** dialog.
28. Press the **OK** button in the **In-Line Fields** dialog.
29. Press the **OK** button in the **Prompts for ?List2** dialog.
30. Press **OK** to close the **Window Formatter**.

The INGREDIENT records are related to the RECIPE records through the RecipeNo field. Therefore, the ING:RecipeNo field must be set equal to the RCP:RecipeNo field whenever a new INGREDIENT record is added. If this were not done, there would be no way of knowing which INGREDIENT records belonged to which RECIPE records. Since the records are being added with the In-Line entry procedure, you will use the embed point created especially for the In-Line entry procedure.

► **To add an embed point for priming a new record**

1. Press the **Embeds** button on the **Procedure Properties** dialog.
2. On the **Embedded Source** dialog look for **In-Line Update On Insert After Record is Primed** in the list of embed points and select **Browse List Box (7)** just below it.
3. Press the **Add** button in the **Embedded Source** dialog.
4. Select **Source** and press the **Select** button.
5. To save the recipe number in the INGREDIENT file enter the following: *ING:RecipeNo = RCP:RecipeNo*
6. From the Editor's Main menu, choose the **File** menu and then choose **Close**.
7. When prompted with Save "In-Line Update On Insert Before Record is Primed (Browse List Box (7))", press the **Yes** button.
8. Press the **Close** button on **Embedded Source** dialog.

Setting Up Parent-Child Relationships

Parent-Child relationships can be defined when the records in two browse list boxes are related. Relationships need to be established only when you want to display a limited set of child records for each parent record selected in another browse list box.

Records can be related when each record contains at least one field with the same unique identifying piece of information. For example, each RECIPE record in the RECIPE file is assigned a unique recipe number when the record is initially created. Any time a new INGREDIENT record is created, this same unique number is added to the INGREDIENT file.

Since there will be only one RECIPE record assigned to each individual recipe number, the RECIPE record is the parent to the multiple INGREDIENT records that may be assigned the same number.

► **To set the RECIPE list box as the parent**

1. Press the **Window** button in the **Procedure Properties** dialog.
2. Edit the first (RECIPE) Browse Box control's **Actions** and select **Parent** in the **List Box Relationship** field.
3. Press the **OK** button in the **Prompts for ?List** dialog.

► **To set the second browse list box as the child**

1. Edit the second (INGREDIENT) Browse Box control's **Actions** and select **Child** in the **List Box Relationship** field.
2. In the **Name of Parent List Box** field select **?List**.

To display only those INGREDIENT records which are related to the RECIPE record currently selected, you must define the filter or range information to be used for displaying the child records.

► **To limit the child list to records for the parent**

1. Press the **Display Orders** button in the **Prompts for ?List2** dialog.
2. Select **By Ingredient** from the **Display Orders** list and press the **Edit** button
3. Click the ellipsis button next to the **Range Limit Field** field.
4. Select **ING:RecipeNo** and press the **Select** button.
5. Select **Single Value** in the **Range Limit Type** field. The **Range Limit Boundary** group will appear.
6. Type *RCP:RecipeNo* in the **Range Limit Value** field. This will limit the INGREDIENT records displayed to only those related to the RECIPE record currently displayed in the first browse list box.
7. Press the **OK** button in the **Display Orders** Format dialog.
8. Complete the same process for the **By Measure** display order.
9. Press the **OK** button in the **Display Orders** dialog.
10. Press the **OK** button in the **Prompts for ?List2** dialog.
11. Press the **OK** button to close the **Window Formatter**.

Making the Window Resizable

When using the resizing feature there are several points to consider when designing the resizable window. The controls above the main list will move with the left side unless there is a space of 10 or more dialog units between them in which case the remaining controls move with the right side of the window. When using the resizing feature with both a parent and child window, there are even more points to consider when designing the window. The controls above the main list still stick to the left side of the window until there is a space of 10 or more dialog units. In addition, the bottoms of both list boxes will line up once the window is resized; therefore you should design your screen with them lined up to begin with.

► **To add the resizing feature to a list box**

1. On the **Procedure Properties** dialog check the **Enable Window Resizing** box.
2. Choose **?List** in the **Parent List Box** field.
3. Check the **Enable Secondary Resizing** box.
4. Choose **?List2** in the **Child List Box** field.
5. From the **Procedure Properties** dialog choose **Window** to open the **Window Formatter**.
6. RIGHT-CLICK the window's title bar and choose **Properties**.
7. Select **Resizable** in the **FrameType** field.
8. Check the **Maximize** box in the **Options** group.
9. Press the **OK** button in the **Window Properties** dialog.
10. Press the **OK** button to close the **Window Formatter**.
11. Press the **OK** button in the **Procedure Properties** dialog.

You have now completed the Power Browse Tutorial exercise. Save, Make and Run the application and test all of the features to make sure that they are functioning correctly.

Working with the Example Application

This chapter explains how to create the Power Popup menus in the example application. The dictionary file and the application file for the example application (with the popup menu code removed) have been copied to the TUTOR subdirectory, which makes it possible to complete the tutorial quickly.

This exercise assumes that the Power Popup templates were installed in the directory **C:\CW\TCTPL**. If not, please be sure to substitute the appropriate drive and directory information in place of those given in the instructions.

Two template extensions are used to add popup menus to an application. The global extension is added to the Global Properties of the application and the local extension must be added to the Procedure Properties of each procedure which will utilize the popup menus.

Using the Global and Local Extensions

Since the application file has already been completed and only those steps necessary to add the popup menus will be covered in this exercise, you need to start by including the global extension in the application. This will add all of the necessary **INCLUDE** statements to the application MAP and to setup the global variable declarations.

Adding the PowerPopupLib Global Extension

You need to start the exercise by opening the file **\CW\TCTPL\TUTOR\PPDEMO.APP**. The **Application Tree** should now be displayed.

► **To add the PowerPopupLib extension**

1. Press the **Global** button in the **Application Tree** dialog and then press the **Extensions** button in the **Global Properties** dialog.
2. Press the **Add** button in the **Edit Extensions** dialog to open the **Select Extension** dialog.
3. Select **PowerPopupLib -ToolCraft Power Popup Global Declarations** from the **ToolCraft_PowerPopup** template class and press the **Select** button.
4. The **PowerPopup Global Declarations** should now be listed in the **Edit Extensions** dialog. Press the **OK** button and then press **OK** again in the **Global Properties** dialog.

Now that the global extension has been added, popup menus can be added to the application, however each procedure which will utilize them must also include the local extension template. This template is what allows you to define each of the menus.

Adding the PowerPopupMenu Local Extension

The **Application Tree** should now be displayed. The **BrowseRecipe** and **UpdateRecipe** procedures will utilize popup menus. You will begin by adding a popup menu to the browse box control in the **BrowseRecipe** procedure. The first step is to add the local extension and then define the menu.

► **To add the PowerPopupMenu extension**

1. Select the **BrowseRecipe** procedure and press the **Properties** button in the **Application Tree** dialog. Then press the **E**xtensions button in the **Procedure Properties** dialog.
2. Press the **A**dd button in the **Edit Extensions** dialog. Select **PowerPopupMenu - ToolCraft Power Popup Menus** from the **ToolCraft_PowerPopup** template class and press the **S**elect button.
3. The **Prompts for PowerPopupMenu** dialog will open. Press the **Controls with Popup Menus** button.

The first menu options that will be added utilize the update buttons which have already been added to the procedure. Since they will be accessed from the popup menu, all three update buttons have been hidden (this is done in the Position properties of the control) to save room in the window.

To add a popup menu, you must first select which control the menu will be activated from. You may add a popup menu to any of the controls in a window, except for a group, string or prompt. These controls do not detect the RIGHT-CLICK event.

► **To add a popup menu to a control**

1. Press the **I**nsert button in the **Controls With Popup Menus** dialog.
2. Select **?List** in the **Control to Create Menu On** field.

Now that the control to add the menu to has been selected, the items to be included for this control must be defined. The three update items (Insert, Change and Delete) will be added first.

► **To add items to a popup menu**

1. Press the **Popup Menu Items** button in the **Controls with Popup Menus** dialog.
2. Press the **I**nsert button in the **Popup Menu Items** dialog.
3. Select **ITEM** in the **Menu Item Type** field.
4. Type **&Insert** in the **Menu Item Text** field. The ampersand signifies that the character following it should be the accelerator (hot) key and will be underlined at runtime.
5. The menu item equate field is optional and will not be used in this case. Select **Execute an Instruction** in the **Action when Pressed** field.
6. Type **POST(EVENT:Accepted,?Insert)** in the **Instruction to Execute** field and press the **O**K button.
7. Press the **I**nsert button to add another item.
8. Select **ITEM** in the **Menu Item Type** field.
9. Type **&Change** in the **Menu Item Text** field.
10. Select **Execute an Instruction** in the **Action when Pressed** field.
11. Type **POST(EVENT:Accepted,?Change)** in the **Instruction to Execute** field and press the **O**K button.
12. Press the **I**nsert button to add another item.

13. Select **ITEM** in the **Menu Item Type** field.
14. Type *&Delete* in the **Menu Item Text** field.
15. Select **Execute an Instruction** in the **Action when Pressed** field.
16. Type *POST(EVENT:Accepted,?Delete)* in the **Instruction to Execute** field and press the **OK** button.
17. Keep pressing the **OK** button until you get back to the **Application Tree**. Once you are there, generate the Application, select the Popup Menu option and then **RIGHT-CLICK** in the browse box to test the popup menu. Once you are sure it is working, exit the program and return to the **Application Tree**.

The demo application had the update items as a submenu of the Update Records main menu item. You now need to go back and make the three menu items that were just added a submenu and then define the Update Records main menu item.

► **To edit the local extension properties**

1. Select the **BrowseRecipe** procedure and press the **Properties** button.
2. Press the **Extensions** button in the **Procedure Properties** dialog.
3. Select **ToolCraft Power Popup Menus** in the **Edit Extensions** dialog and press the **Properties** button.
4. Press the **Controls with Popup Menus** button in the **Prompts for PowerPopupMenu** dialog.

► **To add a submenu to a popup menu**

1. Select **?List** in the **Controls with Popup Menus** dialog and then press the **Edit** button.
2. Press the **Popup Menu Items** button in the **Controls with Popup Menus** dialog.
3. Press the **Insert** button in the **Popup Menu Items** dialog.
4. Select **SUBMENU** in the **Menu Item Type** field and press the **OK** button.
5. Notice that the **SUBMENU** entry is at the bottom of the list. Since update items need to be a submenu, this needs to be moved to the top of the list. Select **SUBMENU** and press the Up Arrow Button until **SUBMENU** is at the top.
6. Press the **Insert** button in the **Popup Menu Items** dialog.
7. Select **ENDSUBMENU** in the **Menu Item Type** field and press the **OK** button. Notice that the update items are now inside of the **SUBMENU** and **ENDSUBMENU** entries.
8. To create the main menu item that will call the newly created submenu, press the **Insert** button.
9. Select **ITEM** in the **Menu Item Type** field.
10. Type *&Update Records* in the **Menu Item Text** field and press the **OK** button. The Action when Pressed field does not need to be entered because the template will automatically know to call the submenu which will be listed immediately below the Update Records item.
11. This item needs to be moved to the top of the list so that **SUBMENU** is below it. Again, use the Up Arrow Button to move the entry to the top of the list.

Press the **OK** button until you get to the **Application Tree**. Generate the application and check the popup menu again. Notice that the menu now has one main menu item and an attached submenu indicated by the arrow head next to the item.

Go back to the **Popup Menu Items** dialog for the **?List** control just as you did the last time you generated the application (remember to go to the **Procedure Properties** window for the **BrowseRecipe** procedure and press the **Extensions** button).

For the Change Filter menu item, the LOC:Filter local data variable (which has already been created) will be changed to utilize the following filter condition which has also been set:

LOC:Filter = " OR INSTRING(CLIP(LOC:Filter),UPPER(RCP:Name),1)

This filter currently does nothing because the local data variable LOC:Filter contains no value, so all records are being displayed. You will now add another menu item with a submenu to allow the user to select from a list of preset filter conditions. Since only one filter condition will be active at any given time, a check mark will be used next to the filter items to indicate which one is currently active.

► **To add menu items with check marks**

1. Press the **Insert** button in the **Popup Menu Items** dialog.
2. Select **ITEM** in the **Menu Item Type** field.
3. Type *Change &Filters* in the **Menu Item Text** field and press the **OK** button.
4. Press the **Insert** button in the **Popup Menu Items** dialog.
5. Select **SUBMENU** in the **Menu Item Type** field and press the **OK** button.
6. Press the **Insert** button in the **Popup Menu Items** dialog.
7. Select **CHECK** in the **Menu Item Type** field.
8. Type *&All Records* in the **Menu Item Text** field.
9. Enter *?AllFilter* in the **Menu Item Equate** field. This is optional, but since you will be referring to this item later in an embed point it is easier to use an equate label rather than the menu item number itself.
10. Select **Execute an Instruction** in the **Action when Pressed** field.
11. Enter *CLEAR(LOC:Filter);POST(EVENT:ScrollTop,?List)* in the **Instruction to Execute** field and press the **OK** button.
12. Press the **Insert** button to add another item.
13. Select **CHECK** in the **Menu Item Type** field.
14. Type *&Chicken Dishes Only* in the **Menu Item Text** field.
15. Enter *?ChickenFilter* in the **Menu Item Equate** field
16. Select **Execute an Instruction** in the **Action when Pressed** field.
17. In the **Instruction to Execute** field, type *LOC:Filter='CHICKEN';POST(EVENT:ScrollTop,?List)* and press the **OK** button. Make sure that you type CHICKEN in upper case letters since the filter condition checks for all upper case letters.
18. Press the **Insert** button to add another item

19. Select **CHECK** in the **Menu Item Type** field.
20. Type *&Beef Dishes Only* in the **Menu Item Text** field.
21. Enter *?BeefFilter* in the **Menu Item Equate** field
22. Select **Execute an Instruction** in the **Action when Pressed** field.
23. Type *LOC:Filter = 'BEEF';POST(EVENT:ScrollTop,?List)* in the **Instruction to Execute** field and press the **OK** button.
24. Press the **Insert** button in the **Popup Menu Items** dialog.
25. Select **ENDSUBMENU** in the **Menu Item Type** field and press the **OK** button.

The change filters submenu is now complete. Notice that the filter condition items are all located inside the submenu structure and that the submenu is directly below the Change Filters main menu item.

The next menu item will call another browse list to display related records. The BrowseIngredients procedure has already been defined with parameters to demonstrate how parameters are passed.

► **To add a menu item that calls a procedure with parameters**

1. Press the **Insert** button in the **Popup Menu Items** dialog.
2. Select **ITEM** in the **Menu Item Type** field.
3. Type *&View Ingredients* in the **Menu Item Text** field.
4. Select **Call a Procedure** in the **Action when Pressed** field.
5. Select **BrowseIngredient** in the **Procedure Name** field.
6. Enter *(RCP:RecipeNo)* in the **Parameters** field and press the **OK** button.

A separator line needs to be added to the menu to separate the Print Recipe item from the other items. A separator is added as a separator type of menu item.

► **To add a separator line in a menu**

1. Press the **Insert** button in the **Popup Menu Items** dialog.
2. Select **SEPARATOR** in the **Menu Item Type** field and press the **OK** button.

The Print Record menu option calls the PrintRecipe procedure. This procedure already has the range value set to print only the currently selected Recipe record so you do not need to use any parameters to call this procedure.

► **To add a menu item that calls a procedure**

1. Press the **Insert** button in the **Popup Menu Items** dialog.
2. Select **ITEM** in the **Menu Item Type** field.
3. Type *&Print Record* in the **Menu Item Text** field.
4. Select **Call a Procedure** in the **Action when Pressed** field.
5. Select **PrintRecipe** in the **Procedure Name** field and press the **OK** button.
6. Keep pressing the **OK** button to return to the **Procedure Properties** dialog.

The menu for the ?List control is now complete. Since the check mark on CHECK type menu items is a toggle and the Change Filter items need to have only the last item selected checked, some source code needs to be added to take care of this.

► **To use embed points to call other popup procedures**

1. Press the **E**mbeds button in the **Procedure Properties** dialog.
2. In the **Embedded Source** dialog, select the **After Popup Menus are Created** embed and press the **A**dd button. (This embed point is just after the **Window Event Handling** embed.)
3. In the **Select Embed Type** dialog, select **SOURCE** and press the **S**elect button.
4. Type the following source code in the editor: *SetPopupChecked(?List,?AllFilter,TRUE)*
5. From the menu, select **F**ile and **C**lose. When prompted with **Save "After Popup Menus are Created"**, press the **Y**es button.
6. Select the **Other Control Event Handling** embed point for the ?List control and then press the **A**dd button.
7. Type the following source code in the editor:

```

CASE EVENT()
OF POPUP:EVENT:Accepted
CASE POPUP:CHOICE
OF ?AllFilter
SetPopupChecked(?List,?AllFilter,TRUE)
SetPopupChecked(?List,?ChickenFilter,FALSE)
SetPopupChecked(?List,?BeefFilter,FALSE)
OF ?ChickenFilter
SetPopupChecked(?List,?ChickenFilter,TRUE)
SetPopupChecked(?List,?AllFilter,FALSE)
SetPopupChecked(?List,?BeefFilter,FALSE)
OF ?BeefFilter
SetPopupChecked(?List,?BeefFilter,TRUE)
SetPopupChecked(?List,?AllFilter,FALSE)
SetPopupChecked(?List,?ChickenFilter,FALSE)
END
END

```

8. From the menu, select **F**ile and **C**lose. When prompted with **Save "Other Control Event Handling (?List)"**, press the **Y**es button.
9. Press the **C**lose button and then press the **O**K button in the **Procedure Properties** dialog to return to the **Application Tree**. Generate the application and test the menu.

To demonstrate the process of running another program, some of the Windows accessory programs: Calculator, Calendar and Notepad will be called. If you don't have these installed on your machine then you can call some other program to test this menu.

This menu will be added to the UpdateRecipe procedure. A region control has already been placed over the entire window. It is very important when defining a region to make sure that it is the last field (use **SetOrders**) in the window. If it is not, it can cause some of the other controls not to function. The first thing you will need to do is add the local extension just as you did for the BrowseRecipe routine.

► **To add a menu item that runs a program**

1. Select the **UpdateRecipe** procedure and press the **Properties** button in the **Application Tree** dialog.
2. Press the **E**xtensions button in the **Procedure Properties** dialog.
3. Press the **A**dd button in the **Edit Extensions** dialog. Select **PowerPopupMenu - ToolCraft Power Popup Menus** from the **ToolCraft_PowerPopup** template class and press the **S**elect button.
4. The **Prompts for PowerPopupMenu** dialog will open. Press the **Controls with Popup Menus** button.
5. Press the **I**nsert button in the **Controls With Popup Menus** dialog.
6. Select **?Region1** in the **Control to Create Menu On** field.
7. Press the **Popup Menu Items** button in the **Controls with Popup Menus** dialog.
8. Press the **I**nsert button in the **Popup Menu Items** dialog.
9. Select **ITEM** in the **Menu Item Type** field.
10. Type **&Calendar** in the **Menu Item Text** field.
11. Select **Run a Program** in the **Action when Pressed** field.
12. Type *Calendar.EXE* in the **Program Name** field and press the **O**K button.
13. Press the **I**nsert button in to add another item.
14. Select **ITEM** in the **Menu Item Type** field.
15. Type *C&alculator* in the **Menu Item Text** field.
16. Select **Run a Program** in the **Action when Pressed** field.
17. Type *Calc.EXE* in the **Program Name** field and press the **O**K button.
18. Press the **I**nsert button in the **Popup Menu Items** dialog.
19. Select **ITEM** in the **Menu Item Type** field.
20. Type *&Notepad* in the **Menu Item Text** field.
21. Select **Run a Program** in the **Action when Pressed** field.
22. Type *Notepad.EXE* in the **Program Name** field and press the **O**K button.
23. Press the **O**K button in the **Popup Menu Items** dialog and then again in the **Controls with Popup Menus** dialog.

Generate the application and update one of the Recipe records in the browser to get to the update Recipe form. Notice that the popup menu cannot be activated when you RIGHT-CLICK on an active control. This is because the control is handling the right mouse click itself and the region never receives this event. You can force the control to pass its right mouse event onto the region control by using the `AliasPopup` function.

► **To use an Alias menu**

1. Press the **I**nsert button in the **Controls With Popup Menus** dialog.
2. Select **?Region1** in the **Control to Create Menu On** field.
3. Check the **Use Alias Popup** check box.
4. Enter *FIRSTFIELD()* in the **Alias Popup Control From** field and *LASTFIELD()* in the **Alias Popup Control Thru** field. Field numbers and field equate labels can be used in both of these fields. The **Alias Popup Control Thru** field is optional.
5. Press the **O**K button and return to the **Application Tree**.

Save, **M**ake and **R**un the application and test all of the popup menus. If they are all working, you have successfully completed the exercise and should now be well on your way to having the basic skills you will need to use Power Popup in all of your own applications.

The **R**eference chapter should be very useful when designing your own applications. Both the global and local extensions are discussed along with all ten of the Power Popup procedures and functions included in the Power Popup Library. You should read through the next chapter now to make sure that you understand all of the Power Popup procedures and functions.