Integrated Development Environment (IDE) Guide

SoftVelocity

SoftVelocity Incorporated
2769 East Atlantic Blvd.
Pompano Beach, Florida 33062
(954) 785-4555
www.softvelocity.com

Printed in the United States of America (1103)

Contents:

Introduction

Welcome to the IDE Guide! This document is a complete reference to the static (non-template) parts of the Clarion Integrated Development Environment, and is to be considered as a close companion to the *Online User's Guide* help document.

Once you've become familiar with the Clarion development environment through the various tutorials included in the Online User's Guide, you can refer back to this document for more complete in-depth information regarding the IDE.

This document covers each major component of the development environment from the Data Dictionary Editor through the Application Generator and Window/Report Formatters.

Please refer to the main help and associated PDFs for complete information on the templates, and the ABC Library Reference for documentation regarding the Application Builder Class Library. Please refer to the Database Drivers help and PDF for complete information on the database drivers that come with this product. Please refer to the Language Reference help topics and PDF for complete information on Clarion language syntax and related examples. And don't forget to make liberal use of the rest of the extensive on-line help topics.

Documentation Conventions

The documentation uses the typeface and keyboard conventions that appear below.

Typeface Conventions

Italics Indicates what to type at the keyboard, such as *Type This*. It also indicates the text of a window's title bar.

ALL CAPS Indicates keystrokes to enter at the keyboard, such as ENTER or ESCAPE, and also indicates Clarion Language keywords. For more information on these keywords, see the *Language Reference* PDF.

Boldface Indicates commands or options from a pull down menu or text in a dialog window.

COURIER NEW Used for diagrams, source code listings, to annotate examples, and for examples of the usage of source statements.

These graphics indicate information that is not immediately evident from the topic explanation.

Keyboard Conventions

F1 Indicates a keystroke. Press and release the F1 key.

ALT+X Indicates a combination of keystrokes. Hold down the alt key and press the x key, then release both keys.

General IDE Support

The following section describes dialogs and support that is generic to all parts of the Clarion Integrated Development Environment (IDE).

General Setup and Navigation

Database Driver Registry Dialog

The Clarion database drivers are pre-registered for you. Should you remove one and need to reinstall it, or if you obtain a new driver, you must use this dialog box to register the new driver. To do so:

1. Choose **Setup ▸ Database Driver Registry**.

2. Press the **Add** button in the **Database Driver Registry** dialog box.

3. Select the file driver, which normally has a DLL file extension, from the **Add Database Driver Open File** dialog, and press **OK**.

The database drivers belong in the ..\BIN subdirectory.

4. Press **OK** to close the **Database Driver Registry** dialog.

The dialog contains these additional buttons:

Remove
Deletes the selected driver from the registry. This does not delete the .DLL file from your drive.

Update
Updates the registry information for the selected driver.

Reminder: when distributing your application, you must ship the database driver library files used by your application. The file names are listed within this dialog.

Dictionary Synchronizer Registry Dialog

This window displays the database drivers that are pre-registered for you to use with the Dictionary Synchronizer. Should you remove one and need to reinstall it, or if you obtain a new driver, you must use this dialog box to register the new driver. To do so:

1. From the main environment menu, choose **Setup ▸ Dictionary Synchronizer Registry**.

2. Press the **Add** button in the **Dictionary Synchronizer Registry** dialog box.

3. Select the file driver, which normally has a DLL file extension, from the **Add Database Driver Open File** dialog, and press **OK**.

The database drivers belong in the ..\BIN subdirectory.

4. Press **OK** to close the **Dictionary Synchronizer Registry** dialog.

The dialog contains these additional buttons:

Remove
Deletes the selected driver from the registry. This does not delete the .DLL file from your drive.

Update
Updates the registry information for the selected driver.

Reminder: when distributing your application, you must ship the database driver library files used by your application. The file names are listed within this dialog.

Editor Options Dialog

To personalize your editing environment, customize appearance and cursor behavior with the **Editor Options** dialog. To view the dialog, choose **Setup ▸ Editor Options**. Select the corresponding tab to set specific Text Editor options.

Font

The Font settings allow you to control the appearance of text displayed in the Text Editor.

Default Uses a default "IBM" font style built in to the run time library.

System Uses the "system" font as set by your operating system. This setting is stored in the C6EE.INI file.

Custom Press the ellipsis button to set a custom font to use in the Text Editor.

Insertion

Indent New Line To automatically give a new line the same indention as the previous line, check this box. This will make your code more readable.

Insert Within Column
When the insertion point is in the middle of a line, ENTER adds a new line after the current line.

Automatic Word-wrap
To cause automatic line breaks at column 70, check this box.

Split Line at Cursor When this box is checked, ENTER will split the current line at the insertion point (cursor). The second part of the line will appear on a new line. When this box is not checked, ENTER inserts a blank line below the current line, *without* splitting the current line.

Tab Size To set the default spacing between tabs, enter a number in the **Tab Size** box.

Block

Automatic Block Delete
To replace the selected text when pasting, check this box. To insert before a selected block, uncheck the box.

Remove Block On Copy
To delete the selected text when copying, check this box.

Colors

These options allow you to set color choices for twenty-one different Clarion language elements. For example, make Clarion keywords appear in red, or make equates appear in green.

Select a language or text element in the **Color Groups** list box, then CLICK on a color selection box. The sample text shows you how the selected language element will appear in the Text Editor.

Color Groups Highlight the language or text element to receive a color assignment.

Color To assign a color to the selected language element, CLICK on a color selection box.

Default To assign the default color to the selected language element, check this box.

Custom To reset the custom color for the selected language element, check this box.

Sample Text Shows how the selected language element will appear in the Text Editor.

Enabled To apply the color syntax highlighting to the file types listed in the **Source Extensions** box, check this box.

Source Extensions
To specify the file types that color syntax highlighting is applied to, type a list of file extensions separated by semicolons.

Restore Defaults To assign the default colors to all language and text elements, check this box.

Saving

Make Backup Files

To cause the Text Editor to make a backup file (.BAK) each time you explicitly save a source file, check this box. The .BAK file contains the source as it was previously saved.

Prompt for Reload if file changed

To receive a "*source*.CLW has changed on disk. Do you want to reload?" message whenever the Text Editor detects such a change, check this box.

Automatic Save time (minutes)

Clarion's Automatic Save option saves the current file according to the time interval you specify. Type the desired number of minutes in this box. The a copy of the last *explicitly* saved version of the file is stored in a temporary file and is used to restore if you cancel the current session and choose not to save.

Extra

Home to First Column

By default, pressing the **Home** key normally moves the cursor to the first text character on the selected line. Check this box to force the cursor to always move to the first column when the **Home** key is pressed.

Show Field Box in Embed Editor

This setting allows the popup field box to display in the Embed editor. This setting does not affect source files accessed outside of the Embed editor.

General Environment Menu Commands

The General Environment Menu refers to the menu commands of Clarion's IDE (Integrated Development Environment) when no special area is active (Dictionary Editor, Application Generator, etc.).

File Menu

New	Opens the **New** dialog, which lets you create a new dictionary or other type of file.
Open	Calls the **Open File** dialog, allowing you to open a Dictionary file.
Pick	Calls the **Pick** dialog, listing the most recently used files in a list box.
Close	Closes the active file.
Quick Start New App	Calls the Clarion Quick Start dialog window. You will first access the New dialog window, to designate the name, and where you want to create it.
Save	Saves the active file.
Save As	Saves the active file under a new name, which you specify.
Save All	Saves all the active files.
Print	Prints the active document.
Print Setup	Calls the **Printer Setup** dialog, allowing you to configure your printer.
Change Directory	Opens a Windows **Directory** dialog. Any subsequent file dialogs will default to the selected directory.
Search Files	Opens the **File Search** dialog. Allows standard searches for files.
Browse Database	Loads the Database Manager

Convert Application

The **Convert Application** command starts the Clarion Application Conversion Wizard to convert applications developed in one Clarion environment to another Clarion environment (newer Clarion version, different templates, etc.). See Application Converter for more information.

Note:

If your templates are unchanged, you can open application files with newer versions of Clarion and the file conversion is automatic. The Application Conversion Wizard carries out more complex conversions.

Exit Quits the program.

Data Modeller

Note:

This menu item is only available in the Enterprise Edition.

Project Menu

Set Calls the **Open File** dialog, allowing you to change the active .APP or .PRJ.

New Calls the **New Project File** dialog, allowing you to create a new project.

Load Opens the **Project Tree** dialog for hand coded projects, or **Application Tree** dialog for generated projects.

Edit Opens the **Project Tree** dialog, allowing you to add or edit component files in the current project.

Make Compiles and links the active application or project, which is named on the caption bar.

Make & Run Compiles, links and runs (if a target executable) the active application or project, which is named on the caption bar.

Run Executes the active application or project, which is named on the caption bar.

Debug Loads the Debugger and prepares the active application or project, listed on the caption bar, for debugging.

Make Statistics Calls the **Make Statistics** dialog. Lets you view a statistical profile of the most recent make.

Auto Make Before Run
Toggles the Project System setting which forces a recompile each time you choose the Run command.

File Save Before Run Toggles the Project System setting, which saves the source code file each time you choose the Run command.

Minimize on Run Toggles the Project System setting which minimizes CW before displaying the application each time you choose the Run command.

Wait for Termination on Run
Toggles the Project System setting, which suspends CW until after you terminate the application upon executing it with the Run command.

Setup Menu

Editor Options Calls the **Editor Options** dialog, which lets you customize the appearance and behavior of the Text Editor.

Edit Redirection File Loads the Redirection File in a document window, ready for editing.

Database Driver Registry
Calls the **Database Driver Registry**, which lets you register database drivers.

Dictionary Synchronizer Registry
Call the **Dictionary Synchronizer Registry**, which lets you register database drivers to use with the Dictionary Synchronizer.

Template Registry Calls the **Template Registry** dialog, which lets you register and manage templates.

Window Formatter Options
Calls the **Window Formatter Options** dialog, which sets the default position and size values applied when auto-populating controls, or when aligning controls with the alignment tools.

Application Options Calls the **Application Options** dialog, which lets you specify default settings for the Application Generator.

Dictionary Synchronizer Options
Calls the **Dictionary Synchronizer Options** dialog, which lets you specify default settings for the Dictionary Synchronizer.

Dictionary Options Calls the **Dictionary Options** dialog, which lets you specify default settings for the Dictionary Editor.

VCS Options Calls the **Configure VCS Interface** dialog, which lets you configure a selected Version Control System for your dictionaries and applications.

Window Menu

Tile Vertically Arranges open document windows side by side in a vertical orientation.

Tile Horizontally Arranges open document windows side by side in a horizontal orientation.

Cascade Arranges open document windows in overlapped fashion so that all caption bars are visible

Arrange Icons Arranges iconized windows along the bottom of the Clarion Application frame.

(Window List) Lists all open document windows by their caption bar text according to the order they were opened. Choosing a window from the list brings the window to the top.

Help Menu

Contents Opens the Windows Help application and displays a list of main topics.

Search for Help On Opens the **Search** dialog in the Windows Help application, allowing you to search for help topics containing a specific keyword.

How to Use Help	Opens the Windows Help application and displays instructions for using the Help system.
About Clarion	Displays the program name, version, registration, and copyright information.

Toolbar Menu

 **Pick**
Opens an existing file by selecting the file from the Pick dialog window.

 **New**

Creates a new file of the type you select in the folder you select.

 **Open**
Opens an existing file by selecting a file from the Windows file open dialog.

 **Save**
Saves the active file to disk.

 **Make**
Generates source code, then compiles and links the active application or project.

 **Run**
Runs the active program after optionally saving and making the project.

 **Debug**
Makes the active project then starts the debugger.

Template Registry Dialog

Template files (*.TPL) drive the Application Generator. Each procedure template contains generic or "model" code. The templates are interactive--they process the information you specify when you design the application within the IDE. Clarion evaluates the template file twice:

- Before creating your application, Clarion pre-processes the template file and stores the information in the *REGISTRY.TRF* file. Pre-processing occurs only when the Application Generator detects a new or changed template.

  When it pre-processes the template file, the Application Generator stores a list of all the information you must provide each procedure. It also determines the points at which you can embed your own Clarion source code to customize a procedure. The registry file contains the default windows, dialogs, menus, report designs, default data, and formulas. In the design process, you customize these defaults.

- At code generation time, the Application Generator evaluates the information you provide in the design process--from the data dictionary, and the .APP file--then processes it along with the template language statements and symbols in the *REGISTRY.TRF* file to generate your source code.

Each template can contain multiple types of procedure templates from which you select to create the procedures in your application. Before you can use a template it must be in the Template Registry.

The **Template Registry** dialog provides command buttons for file maintenance options for the registry:

Note:

The Register, Unregister, Enable, and Disable buttons are not available in Multi-Developer mode. To change this setting, select Setup ▸ Application Options.

Register	Calls the Open File dialog, which allows you to register a template (.TPL) file.
UnRegister	This button deletes the currently highlighted template class from REGISTRY.TRF.
Enable	This button enables the currently highlighted template class or procedure (if you had previously disabled it).
Disable	This button disables the currently highlighted template class or procedure, which makes it unavailable to your application.

Properties This button accesses the **Template Procedure** dialog. Press the Defaults button to edit default global data or structures contained in the procedure template.

Edit Definition This button displays the Template code (the .TPL) in the Text Editor. You may edit the code in this window.

If the currently highlighted item in the Template Registry tree is a module, the text window opens to the first line of the MODULE definition. If a procedure, it opens to the first line of the PROCEDURE. If the highlighted template is one of several in a single file, it opens to the first line of the highlighted template.

Standard Windows File Dialog

This dialog lets you specify a filename and directory for a file that you wish to Create, Open, Save to, or Select. The Title Bar indicates the function for which it is intended.

The Pick List

The **Pick** dialog lists all the most recently used files in a list box categorized by **Application, Dictionary, Project, Database, Source, Report** and **All**. See Core IDE Elements for an expanded explanation of each of these tabs.

Each of these tabs displays a pick list of the most recently used files of that type:

The **Pick** dialog provides the following buttons:

Select	Opens the currently selected file.
Remove	Removes the currently selected file from the Pick list.
New	Lets you create a new file. The **Database** and **All** tab controls will redirect you to the **New** menu, since New is not an option from the Pick List for those areas.
Open	Lets you open a file not on the Pick list.
Cancel	Closes the Pick List without selecting a file or IDE component.
Help	Calls this window.

Core IDE Elements

Clarion's development environment is a program that helps you write other programs. The development environment manages all the files (source code, executables, resources, etc.) and all the processes (editing, compiling, linking, etc.) needed to successfully develop Windows programs. The *Getting Started* and *Learning Clarion* books provide a wonderful general introduction to Clarion's development environment.

The tab controls on the Pick List, and New and Open menu options, redirect you to the core components of the Clarion IDE (Integrated Development Environment):

Application (*.app)
Selecting the Application tab control or menu item, and opening a file with a .APP extension, starts the Application Generator. By convention, a Clarion application refers to a project whose source code is generated by templates and whose instructions for compiling and linking are stored within the application file (.APP), rather than in a separate Project System file (.PRJ).

Dictionary
Selecting the Dictionary tab control or menu item, and opening a file with a .DCT extension, starts the Dictionary Editor. Data Dictionaries are created and maintained here. The Data Dictionary is the central repository for information about your application's tables, columns, keys, and relationships. The information stored includes how and where the data is stored on disk, as well as how the data is presented to end users on reports and computer screen. This will usually be the first place to visit when starting a new project.

Project
By convention, a Clarion project includes some set of source code, plus the instructions (.PRJ file) for compiling and linking to produce an executable program. Project typically refers to a manually coded program rather than a template-generated program. The Clarion Project System visually manages the project information. It maintains tree diagrams of the source files, external libraries, resources, and other project components.

Database
Clarion supports a wide variety of database formats. This tab control provides access to the Database Manager, which in turn provides you to direct access to tables without the need of creating an application. The Database Manager is designed to allow developers free access to their tables.

As you open a particular table, you will need to select the particular database type (i.e., TopSpeed, Pervasive, MS-SQL, ODBC, etc.). The information provided to you in the Database Manager varies with different database drivers.

Source
This tab control accesses the Text Editor. If you allow the Application Generator to write most of your source code, you will probably only use the Text Editor to write your embedded source code (see Embedded Source Dialog). If you write your own source code "from scratch", you will probably use the Text Editor extensively to create and manage your code. In addition, the Project System is integrated into the Text Editor, allowing you to effortlessly generate executables, libraries, and dynamic linked libraries (DLL).

Report
Selecting the Report tab control, and selecting or opening a file with a .TXR extension, starts the Clarion Report Writer. The Clarion Report Writer lets you create sophisticated reports from your data files, without the need of creating an application. See the Report Writer documentation for more information.

All
Selecting this tab provides you with a history of all files that you have accessed in the Clarion IDE. Selecting a file results in a "smart" open, redirecting you to the correct area of the environment.

Edit Picture String Dialog

This dialog lets you quickly choose and customize a picture token. This interface appears throughout the IDE, specifically in the Window Formatter, Report Formatter, and Dictionary Editor.

Picture tokens provide a masking format for displaying and editing variables. Picture tokens may be used as parameters of STRING, ENTRY, or STRING OPTION declarations in SCREEN structures; as a parameter of STRING statements in a REPORT structure; as a parameter of some Clarion procedures and functions; or, the parameter of STRING, CSTRING and PSTRING variable declarations. There are seven types of picture tokens:

Numeric and Currency Pictures
Scientific Notation Pictures
Date Pictures
Time Pictures
Pattern Pictures
Key-in Template Pictures
String Pictures

See the *Language Reference Manual* for more information.

Pool	Select a Pool picture from the drop list.
Save As	Press this button to save the current picture to the Picture Pool
Delete	Press this button to remove the current picture from the Picture Pool
Picture Type	Choose one of the picture token types from the drop down list.
Picture	The actual picture string under construction. You can directly edit the string, or edit it by specifying options in the dynamic controls that appear in the lower part of the dialog box for individual picture token types.

The options are self explanatory, such as **Number of Characters**, **Length**, and **Decimal Places**.

Picture Editor (Edit Picture)

Some controls let you specify a picture token that provides a special format for display or printing of variables

There is a great variety and diversity of picture token syntax that depends on the type of data you format: strings, numbers, currency, scientific, dates, times, etc.

The **Edit Picture** dialog lets you quickly and easily build an appropriate picture token without memorizing picture token syntax. Invoke this easy to use dialog by pressing the ellipsis (...) button beside the **Picture** prompt in the properties dialog.

Example An example of the display format currently specified in the dialog. What you see is what you get.

Pool Select a predefined picture from the drop list.

Save As Press this button to save the displayed picture to the Pool, and name the saved picture. The saved pictures are available in the **Pool** drop-list. You can save your most frequently used pictures, and then quickly reuse them from the Pool.

Tip: The Picture Editor stores the pictures in the ..\BIN\C6Pict.ini file.

Delete Press this button to delete the displayed Pool picture currently displayed in the **Pool** field.

Picture The picture token currently specified. This picture token produces the example shown.

Picture Type Select the type of data to format from this drop-down list. Choose from:
String
Numeric and Currency
Scientific Notation
Date
Time
Pattern
Key-in Template

String

String Pictures specify a length with no other formatting.

Length Specify the length of the string. This length also determines the width of the control if the width is not otherwise specified by the control's AT attribute.

Numeric and Currency

Numeric and Currency picture tokens specify a length, plus conventional formatting to convey positive and negative values, various currencies, etc.

Size The total number of significant digits, plus any formatting characters. For example, $22.25- is 4 significant digits + 3 formatting characters for a size of 7.

Decimal Digits The number of digits to the right of the decimal.

Currency Choose from None, Leading, and Trailing. None shows no currency symbol. Leading puts the currency symbol to the left of the number and Trailing puts the symbol right of the number.

Symbol The currency symbol to display: either a dollar sign ($) or a string constant.

Negative Sign Specify how negative values are formatted.

- **Bracket** Negatives surrounded by parentheses.
- **Leading** Negatives get a leading minus sign.
- **Trailing** Negatives get a trailing minus sign.
- **None** No sign display.

Decimal Separator Specify the character inserted between the integer and fractional portion of the value.

- **Period** Period is the separator.
- **Comma** Comma is the separator.
- **None** Displays no separator.

Grouping Specify the character inserted at every third digit to aid readability.

- **Comma** Comma is the separator.
- **Period** Period is the separator.
- **Space** Space is the separator.
- **Hyphen** Hyphen is the separator.

Leading Character Specify the character to represent leading zeroes.

- **Clip** Remove leading zeroes so that any leading format characters abut the left most digit.
- **Zero** Leading zeroes display as zeroes (0).
- **Space** Leading zeroes display as spaces ( ).
- **Asterisk** Leading zeroes display as asterisks (*).

Blank When Zero Check this box to display nothing when the value is zero.

Scientific

Scientific Notation picture tokens let you display very large or very small numbers with a decimal format raised by a power of ten. The display takes the form -9.99e+999.

Number of Characters The total number of characters, including the 7 format characters. For example, -1.96e+007 requires 10 characters.

Leading Digits The number of digits to the left of the decimal point (typically 1).

Decimal Separator Specify the character inserted at every third digit to aid readability.

Point Period is the separator.

Comma Comma is the separator.

Space Space is the separator.

Separator Specify the character inserted between the integer and fractional portion of the value.

Point Period is the separator.

Comma Comma is the separator.

Blank When Zero Check this box to display nothing when the value is zero.

Date

Date picture tokens let you display dates in a number of different formats. Choose the format you want from the **Format** drop-down list.

Better still, date picture tokens in entry fields automatically invoke Clarion's run-time date parsing functions, so you can enter '21' and Clarion expands it to the 21st day of the current month and year. Or you can enter 'DEC' and Clarion expands it to the 1st day of December of the current year. The date is then formatted according to the picture token.

Tip

The MASK attribute (Entry Patterns check box) on a window preempts the date parsing functions.

Format Choose the format you want from the drop-down list. What you see is what you get except for the Windows Short and Windows Long formats. Additionally, the separator character and leading zeroes may be specified independent of the chosen format.

Windows Short

Uses the short date format specified in the Windows control panel or the Windows 95 Regional Settings control panel.

Windows Long

Uses the long date format specified in the Windows control panel or the Windows 95 Regional Settings control panel.

Separator Choose from Standard (/), Period (.), Dash (-), Space ( ), and Comma (,).

Leading Characters Specify the character to represent leading zeroes.

Zero Leading zeroes display as zeroes (0).

Blank Remove leading zeroes.

Asterisk Leading zeroes display as asterisks (*).

Two digit date range Change the default century interpretation for dates input with a two digit year. By default, Clarion assumes any date input with a two digit year (i.e. the century value is omitted) falls between today-**80** years and today+19 years.

For example, if today is June 1, 1996 and the date input is 9/2/59, Clarion assumes the 59 means 1959 since 1959 falls between today-80 years (June 1, 1916) and today+19 years (June 1, 2015). To force a different interpretation, set the **Two digit date range** to 30. Now Clarion assumes the 59 means 2059 since 2059 falls between today-30 years (June 1, 1966) and today+69 years (June 1, 2065).

Blank When Zero Check this box to display nothing when the value is zero.

Time

Time picture tokens let you display times in a number of formats. Choose the format you want from the **Format** drop-down list.

Format Choose the format you want from the drop-down list. What you see is what you get except for the Windows Short and Windows Long formats. Additionally, the separator character and leading zeroes may be specified independent of the chosen format.

Windows Short

Uses the time format specified in the Windows control panel.

Windows Long

Uses the time format specified in the Windows control panel.

Separator Choose from Standard (:), Period (.), Dash (-), Space ( ), and Comma (,).

Leading Character Specify the character to represent leading zeroes.

- **Zero** Leading zeroes display as zeroes (0).
- **Blank** Remove leading zeroes.

Blank When Zero Check this box to display nothing when the value is zero.

Pattern

Pattern pictures let you build custom display formats for various numbers: phone numbers, social security numbers, room numbers, dates, times, measurements, etc.

Picture Type the picture token between the 'P's according to the Legend below. Your picture token can include any displayable characters, including all the standard keyboard characters.

At runtime, the constants in the picture token display just as they appear in the token. The left angle (<) and the pound sign (#) resolve into the individual digits from the display variable.

Legend
< integer, blank if zero
integer
constant (any displayable char except < and #)

Blank When Zero Check this box to display nothing when the value is zero.

Tip

To use a lowercase p in your picture, use an uppercase P at the start and end of your picture token. To use an uppercase P in your picture, use a lowercase p at the start and end of your picture token.

Key-in Template

Key-in Pictures let you build custom edit formats for runtime fields containing mixed alphanumeric characters. Although Key-in tokens affect output as well as input, their primary purpose is to provide custom field editing and validation on input.

Picture
Type the picture token between the 'K's according to the Legend below. Your picture token can include any characters (displayable or not), including all the standard keyboard characters.

Legend
< accept an integer, blank if zero
accept an integer
? accept any character (even non-display)
^ accept an upper case character
_ accept a lower case character
| input may stop here
constant (any displayable char except <#?^_| or \)
\ display next char (lets you display <#?^_| or \)

Only alphabetic characters
Check this box to accept only alphabetic characters.

Blank When Zero
Check this box to display nothing when the value is zero.

Tip

To use a lowercase k in your picture, use an uppercase K at the start and end of your picture token. To use an uppercase K in your picture, use a lowercase k at the start and end of your picture token.

New Structure dialog

This dialog lists the default Window and Report Structures contained in DEFAULTS.CLW. You may edit this file to modify these structures or add new window or report types.

The following structures are in the shipping version of DEFAULTS.CLW:

With OK & Cancel
With OK & Cancel - STD:Actions
With OK & Cancel & Help
With OK & Cancel & Help - STD:Actions
With OK & Cancel ICONS
With OK & Cancel ICONS - STD:Actions
With OK & Cancel & Help ICONS
With OK & Cancel & Help ICONS - STD:Actions
Progress Window
Progress Window ICON
Progress Window ICON II
Progress Window ICON III
Progress Window Vertical ICON
Application Main MDI Frame
MDI Child With OK & Cancel
MDI Child With OK & Cancel ICONS
MDI Child Window
Simple Window
System Resizable Window

Reports:
(Paper size A4 - Portrait)
(Paper size A4 - Landscape)
(Paper size Legal - Portrait)
(Paper size Legal - Landscape)
(Paper size Letter - Landscape)
(Paper size Letter - Portrait)

Property Editor

This utility allows you to set properties on one or more controls throughout the IDE. In a window, you can set the tab-stop order of the controls in the window by reordering a list of controls.

After you make changes, press the **Apply** button and you changes are implemented on the underlying window. (You can move the Property Editor out of the way to see the effect.)

The default view is the PropertyTree. This shows an expanded view of all controls of the target structure and their order. Pressing **-** and ¯ buttons move the selected control or branch up or down within the list.

Tip **Moving controls that have controls branched underneath them moves everything in that branch. This makes it easy to move large number of grouped controls.**

You can select multiple controls by simply using the standard Windows select commands. Use Ctrl-Click (selects or tags items you are clicking on, in any order) or by Shift-Click (selects the first and last item and all items between them). With the keyboard, Shift-Up/Down arrow selects items one at a time inclusively and Shift-PgUp/PgDown selects pages of items.

When more than one control is selected, the property editor shows the values for the last control. Values of control properties that are different display in italics.

Tip **You can select multiple controls in the window and report formatters and then press F12. These controls will be selected when the Property Editor opens.**

Conversely, you can select controls in the Property Editor and when you close it, the controls will remain selected in the appropriate Formatter.

There are two views. From the drop list, choose "Property column". The right pane empties so that you may add properties that you wish to affect. Press the "fields to display" button to choose which properties you wish to add to the right pane. They display as resizable columns.

Right-click to select the sort order of this list. For example, you may wish to set all entry controls to have a background color. If you sort by "type", then you may change all the entry controls easily. You may sort by any column visible in the list.

Tip

This dialog is useful for moving controls among overlapping TAB controls on a SHEET or moving a control off a TAB and onto the WINDOW.

Select Font Dialog

Lets you change the font, style (such as bold and italic), font size, color, and font effects (underline and strikeout) for the selected control or window. As you choose options, the dialog box displays a sample of the formatting.

Choose from the following:

Font Type or select a font (typeface) name. The dialog lists the fonts available with the current printer driver and additional fonts installed in your system.

Font Style Select a style. To use the default type style for a given font, select Regular. Depending on the fonts installed, you can choose bold, italic, or bold italic.

Size Type or select a size. The sizes available depend on the printer and the selected font.

Effects Select the formatting options you want. Choose:

Underline - underlines all characters, including the spaces between words, with a single line.

Strikeout - draws a line through selected text.

Color
Type or select one of the sixteen predefined colors. To display color, you must have a color monitor; to print color, you must have a color printer.

Version Control System Support

Version Control System (VCS) support has now been added to the Clarion IDE.

The VCS support in Clarion 6 is different to previous versions of Clarion in that it supports *any* VCS system that has a command line interface. Regarding your applications, It also works on a module level rather than a procedure basis, which is more consistent with general VCS systems.

The VCS support works by checking in and out Dictionary Text (TXD) files for dictionaries and Application Text (TXA) files for applications. One TXA file is created for each module in the application. There is also one TXA for the application's global options.

Clarion 6 comes preconfigured to use RCS. It also comes with default settings to use CVS and MS Visual Source Safe. You can add your own set of Version Control commands, and save it to the list of default sets.

To configure the VCS interface select **Setup ▸ VCS Options**. This will take you to the Configure VCS Interface window. This window is also accessible from the Configure VCS button when doing a Check In or Check Out.

Configure VCS Window

The *Configure VCS Interface* window allows you to set the command line syntax for **check**ing **in**, **add**ing, **delete**s, and **check**ing **out** dictionaries and application modules from the VCS (Version Control System). It also allows you to specify different command lines options depending on whether comments are supplied or not.

You can load pre-defined command lines using the **Load Commands** button. You can save your own command lines using the **Save Commands** interface. You can also delete previously saved commands pressing the **Delete Saved Command Set** button.

VCS command sets are stored in the C60EE.INI file as follows:

```
[VCSSystems]
count=2
System1=RCS
System2=CVS
 [VCSSystem RCS]
Check InNoComment=RCS ci "%F"
Check InHasComments=1
Check InShortComment=RCS ci -m"%S" "%F"
Check InHasLongComments=0
Check InLongComment=
Check InHasBothComments=0
Check InBothComments=
AddNoComment=RCS ci -i -t-"" "%F"
AddHasComments=1
AddShortComment=RCS ci -i -t-"%S" "%F"
AddHasLongComments=1
AddLongComment=RCS ci -i -t"%L" "%F"
AddHasBothComments=0
AddBothComments=
DeleteNoComment=RCS delete "%F"
DeleteHasComments=1
DeleteShortComment=RCS delete "%F" -m "%S"
DeleteHasLongComments=1
DeleteLongComment=RCS delete "%F" -F"%L"
DeleteHasBothComments=0
DeleteBothComments=
Check Out=RCS co "%F"
!more commands follow here
```

If you wish, you can set the extension for the TXA and TXD version files, overriding the default extensions of *APV* and *DCV*. You can also specify the **Base Archive Directory** where the APV and DCV files are created in. The default directory for APV files and DCV files is

<base directory>\<app or dct name>

This directory can be overwritten when checking in or out an application module or dictionary.

Check In Application

To check in parts of an application (APP) to the Version Control System (VCS) you need to have the application open. Next, select **File ▸ Check In** (or press ALT + K). You then will be presented with a list of modules that make up the application. In the list of modules the **Task** column will default to the following:

Add
Modules that have not previously been added to the VCS.
Check In
Modules that have been checked out
Delete
Modules that have previously been checked in to the VCS, but are no longer part of the application.
No Action
Modules that have not been checked out from the VCS.

You can specify **Short** and **Long Descriptions** (VCS Comments) for each component of the application. These prompts will only be enabled if your VCS support them.

To change the **Task** to be performed, either RIGHT CLICK or DOUBLE CLICK on an entry in the list.

The **Project Name** is an application specific entry used when expanding the command line. It is possible to manage several applications within the same Project Name.

The **Archive Directory** is the directory the ACV files (Application Version files as defined in the Configure VCS dialog) will be created in. By default, all application VCS specific information is stored in an AVC file in the same directory as the .APP file.

Press the **Configure VCS** button at any time if you need to modify the Check Out command syntax.

Check Out Application

To check out parts of an application (APP) from the Version Control System (VCS), you need to first open the application. Next, select **File ▸ Check Out** (or press ALT + U).

In this dialog window, select which module parts of the application that you want to check out. The **Task** column will default to *No Action* if you already have the component checked out, and default to *Check Out* otherwise. You can also set the Task to *Import Only* to handle the case where you have checked out the component using the VCS software directly and simply want to import the component into your application.

To change the Task to be performed, either RIGHT-CLICK or DOUBLE-CLICK on an entry in the list.

Press the **Configure VCS** button at any time if you need to modify the Check Out command syntax.

Check In Dictionary

To check in a Dictionary (DCT) to the VCS (Version Control System) you need to have the Dictionary (DCT) open. Next, select **File ▸ Check In** (or press ALT+K) to open this dialog window. You can then decide whether to **Check In**, **Add**, or **Delete** this dictionary from the VCS. The “Check In” state will default to **Add** if the dictionary has not previously been added to the VCS, or default to **Check In** otherwise.

You can specify **Short** and **Long Descriptions** (VCS Comments) for the dictionary. The appropriate comment line will only be enabled if your VCS support them.

The **Project Name** is a dictionary specific entry used when expanding the command line. It is possible to manage several dictionaries within the same Project Name.

The **Archive Directory** is the directory the DCV files (Dictionary Version files as defined in the Configure VCS dialog) will be created in. By default, all dictionary VCS specific information is stored in an DVC file in the same directory as the DCT file.

Press the **Configure VCS** button at any time if you need to modify the Check In command syntax.

Check Out Dictionary

To check out a Dictionary (DCT) from the VCS (Version Control System), you need to have the Dictionary (DCT) open. Next, select **File ▸ Check Out** (or press ALT+U).

Select the **Import Only** check box to handle the case where you have already checked out the dictionary using the VCS software directly and simply want to import the dictionary into the Dictionary Editor.

Press the **Configure VCS** button at any time if you need to modify the Check Out command syntax.

Application Generator

Application Generator Menu Commands

File Edit Application Procedure Project Setup Window Help

Toolbar Buttons

The Application Generator generates your application's code based on the predefined templates you choose from the template registry.

The following lists the menu commands available from the Application Generator. Many dialogs also have Help buttons which you can press to view a help topic specifically about that dialog (the F1 key calls the same topic when the dialog is open).

Note that some of the commands, most notably on the Project and Setup menus, do not specifically reference Application Generator functions. Because the Project System and the Registries are always active, their menu commands are always available.

Popup (right-click) Menu

Properties	Calls the selected procedure's **Procedure Properties** dialog.
Tables	Calls the **Table Schematic Definition** dialog to manage files and keys for the selected procedure.
Window	Calls the **Window Formatter** to visually design a window for the selected procedure.
Report	Calls the **Report Formatter** to visually design a report for the selected procedure.
Data	Calls the **Local Data** dialog to manage memory variables local to the selected procedure. Press the **Properties** or **Insert** button to define or modify variables using the Data Dictionary's **Field Properties** dialog.
Embeds	Calls the **Embedded Source** dialog to manage embedded source code for the selected procedure.
Formulas	Calls the **Formulas** dialog or the **Formula Editor** dialog to manage formulas for the selected procedure.
Extensions	Calls the **Extension and Control Templates** dialog to manage template generated code added to the selected procedure.

Source Opens the Embeditor, which lets you embed your own source code within the context of the surrounding generated code.

Module Calls the **Text Editor** to edit or display the generated source for the selected procedure.

Tip

The source is as last *generated*. Any changes made directly to the generated source code are overwritten by subsequent source code generation.

Synchronize Applies the control attributes specified in the Data Dictionary to all the controls in the selected procedure. The attributes are applied as specified in the **Synchronization** tab of the **Application Options** dialog.

Delete Deletes the selected procedure, leaving it as a ToDo item in your Application Tree. To remove it completely, remove the reference or statement that calls the procedure.

Rename Renames the selected procedure to a name you specify.

File Menu

New Opens the **New** dialog, which lets you create a new application file, dictionary file, source file, or other type of file. You cannot create a new .APP file until you close the current one. You may invoke the **Quick Start Wizard** to help create a new .APP.

Open Calls the **Open File** dialog, allowing you to open another application, dictionary, source or other file (you must first close the current .APP before opening another).

Pick Calls the **Pick** dialog, listing the most recently used files by category.

Close Closes the active file.

Quick Start New App Calls the Clarion Quick Start dialog window. You will first access the New dialog window, to designate the name, and where you want to create it.

Save Saves the active file.

Save As Saves the active file under a new name which you specify.

Save All Saves all the open files.

Print Prints the active document file.

Print Setup Calls the Printer Setup dialog, allowing you to configure your printer.

Change Directory Opens a Windows **Directory** dialog. Any subsequent file dialogs will default to the selected directory.

Search Files Opens the **File Search** dialog. Allows standard searches for files.

Import From Application

Imports a procedure from another .APP file. Select the file from the **Open File** dialog. Then choose procedures to import from the **Select Items to Import** dialog.

You can select an item by DOUBLE-CLICKING on it. A check mark appears to indicate the item is selected. Select additional items by DOUBLE-CLICKING. De-select an item by DOUBLE-CLICKING a previously selected item. Note: Both applications must use the same dictionary.

Import Text Imports the procedures defined in a .TXA (text) file. Create .TXA files with the **Export Text** or **Selective Export** commands.

Export Text Creates a .TXA (text) file from the current application.

Selective Export Creates a .TXA (text) file containing only selected procedures.

Export Project File… Creates a .PRJ (project) file containing the project information of the active application.

Check In Allows you to check in selected application modules to your target Version Control System.

Check Out Allows you to check out selected application modules from your target Version Control System.

Browse Database Browses and edits a database file defined in the current dictionary. Select the file from the **Pick** file dialog, or the **Open File** dialog after specifying a database driver.

Convert Application

The **Convert Application** command starts the Clarion Application Conversion Wizard to convert applications developed in one Clarion environment to another Clarion environment (newer Clarion version, different templates, etc.). See *Application Converter* for more information.

Note:

If your templates are unchanged, you can open application files with newer versions of Clarion and the file conversion is automatic. The Application Conversion Wizard carries out more complex conversions.

Exit Quits the program.

Edit Menu

Properties Calls the selected procedure's **Procedure Properties** dialog. Equivalent to the **Properties** button.

Window Calls the **Window Formatter** to visually design a window for the selected procedure.

Report Calls the **Report Formatter** to visually design a report for the selected procedure.

Data Calls the **Local Data** dialog to manage memory variables local to the selected procedure. Press the **Properties** or **Insert** button to define or modify variables using the Data Dictionary's **Field Properties** dialog.

Embeds Calls the **Embedded Source** dialog to manage embedded source code for the selected procedure.

Extensions Calls the **Extension and Control Templates** dialog to manage template generated code added to the selected procedure.

Source Calls the **Text Editor** to edit or display the generated source for the selected procedure.

Tip

The source is as last *generated*. Any changes made directly to the generated source code are overwritten by subsequent source code generation.

Find Lets you search for a procedure by name. This can be very useful in a large application with dozens of procedures. Type a string to search for in the **Search for Procedure** dialog.

Find Next Lets you search for another procedure, using the same search string as the previous search. If you did not search previously, the **Search for Procedure** dialog appears.

Edit by Name Lets you type the name of a procedure in the **Edit Procedure by Name** dialog, then opens the **Procedure Properties** dialog of the procedure you typed in. This can be very useful in a large application with many procedures.

Delete Deletes the selected procedure, leaving it as a ToDo item in your Application Tree. To remove it completely, remove the reference or statement that calls the procedure.

Application Menu

Properties Displays the **Application Properties** dialog for specifying changes to the .APP file.

Global Properties Displays the **Global Properties** dialog. Equivalent to using the **Global** button in the Application Tree dialog. Set file handling and other application defaults.

Change Dictionary Lets you name a new data dictionary for the application. Type a file name in the **Select New Dictionary** dialog, or press the ellipsis (...) button to choose a new dictionary file from the **Open File** dialog.

If your procedures already reference fields in one dictionary, the Application Generator can only match fields from the new dictionary if both the FILE structure prefix and the RECORD fields are exactly the same. The **New Dictionary** dialog provides a warning message.

View Dictionary Opens the Dictionary Toolbox for the application's data dictionary. The Dictionary toolbox provides a hierarchical list representing your database. This list shows database files, keys, key components, fields, and relationships in an expanding hierarchical tree. The Dictionary toolbox can update all the *existing* items in the application's data dictionary except the dictionary properties.

Insert Module Specifies a new MODULE for generated source code. You can also specify an external .LIB or .OBJ file to add to the project

Template Utility Calls add-in utilities, including CW Wizards from the **Select Utility** dialog. Write your own or install third-party utilities. A simple example is provided in the Template Language Reference.

Synchronize Applies the control attributes specified in the Data Dictionary to all the controls in the application. The attributes are applied as specified in the **Synchronization** tab of the **Application Options** dialog.

Refresh Select this item to force a refresh of the Property Editor window pane.

Redistribute Procedures
Lets you change the number of procedures per module. Specify the new number in the **Select Procedures per Module** dialog. The Application Generator then redistributes the procedures among the modules, according to the new procedures per module number.

Repopulate Modules Lets you change the number of procedures per module, but still keep related procedures together in the same module. Specify the new number in the **Select Procedures per Module** dialog. The Application Generator then redistributes the procedures among the modules, according to the new procedures per module number. Your application may execute slightly faster if you group procedures which commonly execute together in the same module

Renumber Modules Renumbers the modules created by the Application Generator. This is useful for large projects from which procedures have been deleted.

Delete Empty Modules
Removes empty generated source code modules from the project. This is useful for large projects from which procedures have been deleted.

Delete Empty Libraries
Removes empty external source code modules, .LIB files, and .OBJ files from the project. This is useful for large projects from which procedures have been deleted.

Tip

The Delete Modules and Delete Libraries commands do not delete disk files, they simply remove any reference to the files from your application (.app) and project (.prj) files.

Procedure Menu

New Adds a procedure not connected to the procedure tree.

Rename Lets you change the name of the selected procedure. Type a new name in the **Rename** dialog box. Don't forget to change the calling statement too.

Copy Copies the selected procedure to a new procedure, which you name.

Synchronize Applies the control attributes specified in the Data Dictionary to all the controls in the selected procedure. The attributes are applied as specified in the **Synchronization** tab of the **Application Options** dialog.

Refresh Select this item to force a refresh of the Property Editor window pane for a selected procedure.

Change Module Lets you move the selected procedure from one source module to another. Select the destination in the **Select Destination Module** dialog. Your application may execute slightly faster if you group procedures which commonly execute together in the same module.

Change Template Type
Lets you change the procedure type for the selected procedure. Select a new procedure template in the **Select Procedure Type** dialog.

Note:

Changing procedure types can result in "orphaned" embed code. Be careful when using this function.

Project Menu

Set Calls the **Open File** dialog, allowing you to change the active .APP or .PRJ.

New Calls the **New Project File** dialog, allowing you to create a new project.

Load Opens the **Project Tree** dialog for hand coded projects, or **Application Tree** dialog for generated projects.

Edit Opens the **Project Tree** dialog, allowing you to add or edit component files in the current project.

Make Compiles and links the active application or project, which is named on the caption bar.

Make & Run Compiles, links and runs (if a target executable) the active application or project, which is named on the caption bar.

Run Executes the active application or project, which is named on the caption bar.

Debug Loads the Debugger and prepares the active application or project, listed on the caption bar, for debugging.

Make Statistics Calls the **Make Statistics** dialog. Lets you view a statistical profile of the most recent make.

Auto Make Before Run
Toggles the Project System setting which forces a recompile each time you choose the Run command.

File Save Before Run Toggles the Project System setting, which saves the source code file each time you choose the Run command.

Minimize on Run Toggles the Project System setting which minimizes CW before displaying the application each time you choose the Run command.

Wait for Termination on Run
Toggles the Project System setting that suspends CW until after you terminate the application upon executing it with the Run command.

Generate Generates code for all modules that have changed since last code generation.

Generate All Generates code for all modules.

Note:

If you use the DOS command line, or File Manager to delete one of the .CLW files in the current project, please execute this command to regenerate the file. When executing a Make, the Application Generator attempts to regenerate only those source code files which were changed within the Application Generator.

Properties Opens the **Project Tree** dialog, allowing you to add or edit component files in the current project.

Setup Menu

Edit Redirection File Loads the Redirection File in a document window of the Text Editor, ready for editing.

Database Driver Registry
Calls the **Database Driver Registry**, which lets you register database drivers.

Dictionary Synchronizer Registry
Call the **Dictionary Synchronizer Registry**, which lets you register database drivers to use with the Dictionary Synchronizer.

Template Registry Calls the **Template Registry** dialog, which lets you register and manage templates.

Window Formatter Options
Calls the **Window Formatter Options** dialog, which sets the default position and size values applied when auto-populating controls, or when aligning controls with the alignment tools.

Application Options Calls the **Application Options** dialog, which lets you specify default settings for the Application Generator.

Dictionary Synchronizer Options
Calls the **Dictionary Synchronizer Options** dialog, which lets you specify default settings for the Dictionary Synchronizer.

Dictionary Options Calls the **Dictionary Options** dialog, which lets you specify default settings for the Dictionary Editor.

Editor Options Calls the **Editor Options** dialog, which lets you customize the appearance and behavior of the Text Editor.

VCS Options Calls the **Configure VCS Interface** dialog, which lets you configure a selected Version Control System for your dictionaries and applications.

Window Menu

Tile Vertically Arranges open document windows side by side in a vertical orientation.

Tile Horizontally Arranges open document windows side by side in a horizontal orientation.

Cascade Arranges open document windows in overlapped fashion so that all caption bars are visible

Arrange Icons Arranges iconized windows along the bottom of the Clarion Application frame.

(Window List) Lists all open document windows by their caption bar text according to the order they were opened. Choosing a window from the list brings the window to the top.

Help Menu

Contents	Opens the Windows Help application and displays a list of main topics.
Search for Help On	Opens the **Search** dialog in the Windows Help application, allowing you to search for help topics containing a specific keyword.
How to Use Help	Opens the Windows Help application and displays instructions for using the Help system.
About Clarion	Displays the program name, version, registration, and copyright information.

Application Generator Toolbar Buttons

The Application Generator displays a number of toolbar buttons to allow quick access to a variety of fundamental features. Most of these button functions are also accessible using standard menu or keyboard commands. Some of these buttons are also available in other areas of the Clarion IDE.

Pick

Opens an existing file by selecting the file from the Pick dialog window.

New

Creates a new file of the type you select in the folder you select.

Open

Opens an existing file by selecting a file from the Windows file open dialog.

Save

Saves the active file to disk.

Make

Generates source code, then compiles and links the active application or project.

Run

Runs the active program after optionally saving and making the project.

Debug

Makes the active project then starts the debugger.

Generate Module

Generates source code for the currently selected application module.

View Dictionary

Opens the Dictionary Toolbox for the application's data dictionary.

Save and Close

Closes the active file and automatically saves any changes.

Close Without Saving

Closes the active file without saving any changes.

Global

Displays the Global Properties dialog for specifying template based changes to the application.

Project

Open the active project tree window.

Properties

Context sensitive Properties dialog for the currently selected item.

Help

Equivalent to pressing the F1 key and calls the active help topic.

Application Options Dialog

The **Application Options** dialog allows you to specify default settings for each new application you create.

Application

Require Dictionary This options specifies that each new application *must* have a data dictionary.

Default Dictionary Names a data dictionary file as the default which appears in each new Application Properties Dialog. You can change to another dictionary before closing the dialog.

Display Repeated Functions
Check this box to have the Application Generator (Procedure tab only) provide full expansion of procedures called from more than one place in the Application Tree. Clear the box to provide full expansion of only the first instance--every other instance is marked "Expanded Above."

Procedures per Module
The **Procedures per Module** spin control specifies the number of procedures that the Application Generator writes to each source code module. This can affect compile time when used with Conditional Generation turned on. Specifying one procedure per module, for example, means that each successive compile rebuilds only those procedures changed since the last one, and no more. The down side to this is that it requires more disk space. When using the ABC templates, we recommend 10-20 procedures per module.

Application Wizard Check this box to specify the default when creating a new application is to use the Application Wizard. You can override this choice when creating an application by checking or unchecking the Application Wizard box in the Application Properties dialog.

Procedure Wizards Check this box to specify the default when creating a new procedure is to use the appropriate Procedure Wizard. You can override this choice when creating a procedure by checking or unchecking the **Procedure Wizard** box in the **Select Procedure Type** dialog.

Open Dictionary Files and Template Registry as Read-Only
Specifies file management options for multiple developer projects. When checked, Dictionary files and the registry are opened read-only.

Translate controls to control templates when populating
Check this box to have the Window Formatter prompt you with a list of like control templates whenever you place a control. We recommend this setting for new Clarion users.

Import name clash action
Specifies how the Application Generator handles procedure names from an imported application file that clash with procedure names already resident. Choose from the following:

Query on first clash
When the first clash is encountered, the Application Generator prompts for specific instructions on how to handle this clash and each subsequent clash. Choose from **Auto Rename**, **Replace All**, or **Prompt**.

Auto Rename renames all imported procedures with name clashes by appending a sequence number to the imported procedure name.

Replace All replaces all resident procedures with imported procedures of the same name.

Prompt asks for specific instructions for each clash encountered. Choose from **Replace** or **Rename**.

Replace replaces the resident procedure with the imported procedure of the same name.

Rename prompts you to rename the imported procedure.

Ask for alternative
For all procedures with the same name, the Application Generator prompts you to rename each imported procedure.

Auto Rename For all procedures with the same name, the Application Generator renames the imported procedure by appending a sequence number to the name.

Replace previous
For all procedures with the same name, the Application Generator replaces the resident procedures with the imported procedures.

Disable Column Prompts
Specifies that template-generated field-specific prompts will not display. This does not disable prompts created by Control Templates.

Sort Embeds Alphabetically
Check this box to show embed points in alphabetical order in the **Embedded Source** dialog. Clear the box to show the embed points in "logical" order (order of execution). See Embedded Source Code for more information.

Action for Legacy embeds
Specify how the Application Generator handles legacy embed points. Legacy embed points are generally provided for backward compatibility among template chains. They allow newer template chains to conditionally support embed points from older template chains. See LEGACY in the *Template Guide*. Choose from:

Ignore all Application Generator neither displays Legacy embed points at design time nor generates any code embedded therein. We recommend this setting to reduce "clutter" when developing new applications.

Show all and generate

Application Generator displays all Legacy embed points at design time and generates any code embedded therein. We generally do not recommend this setting; however, it can be useful for developers that are very comfortable with a particular template chain and its embed points.

Show filled and generate

Application Generator displays only filled Legacy embed points at design time and generates any code embedded therein. We recommend this setting for applications ported to a new template chain.

Edit embedded source errors in generated code

Check this box to allow you to edit embedded source errors in the generated source code instead of the standard Embeditor. The advantage of this is a logical view of all source code as generated. You can edit the source error and the application generator will not regenerate code until after the Make and Run is completed. This is useful during the testing phase of your development for faster prototyping.

Registry

Template Language code can be logically split among many files. Clarion uses the files to produce one logical template set for creating applications. The Registry Options are mainly for programmers who produce their own template files or make modifications to the default templates.

Re-register When changed

To automatically re-register your templates when the Application Generator detects a change, check the **Re-register When changed** box. This defaults to "On."

Update Template Chain when edited

To automatically update the Template files when making a change in the Template Registry, check the **Update Template Chain when edited** box.

Regenerate Deleted Templates

To specify the Application Generator should re-generate the .TPL and .TPW files from REGISTRY.TRF, should the files be deleted, check the **Regenerate Deleted Templates** box.

#APPLICATION template

Select the default APPLICATION template from the drop-down list. The APPLICATION template controls source code generation. See *Template Overview* in the *Template Guide* for more information.

Generation

Conditional Generation

This check box specifies that only source code modules changed since the last make should be compiled.

Debug Generation

Specifies a text file for the Application Generator to log events to, and turns logging on and off. In case of a fatal error by the Application Generator, this log provides a trace to identify where the problem occurred. You can specify the file name in the **Debug Filename** box.

Generation Message

Allows you to specify what displays during generation. Displaying messages increases generation time slightly, but offers more information on generation progress and more opportunities to cancel.

The choices are:
No Messages -- displays no messages.
Module Names -- displays the Module Name as it generates
Module and Procedure Names -- displays the Module Name as it generates and the Procedure Name as it generates
All Messages -- displays the Module Name as it generates, the Procedure Name as it generates, and each portion of the procedure as it generates.

Enable #ASSERT checking

Check this box to enforce heightened error checking during source code generation. This allows the Application Generator to identify certain template execution problems and notify you during source code generation. This slows the code generation process slightly. Clear the box for faster, but riskier source code generation. See #ASSERT for more information.

Create local maps

Check this box to generate a MAP structure for each source module that prototypes only the procedures referenced in the module. This results in faster compiles whenever you add new procedures or change procedure prototypes, because only the affected modules are recompiled. To generate accurate local maps, you must use the **Procedures** button in the **Procedure Properties** dialog to identify any procedures referenced in embedded source code.

Clear this box to generate a single MAP for the PROGRAM module that prototypes all the program's procedures globally. This results in slower compile times whenever you add new procedures or change procedure prototypes, because the change to the PROGRAM module forces a recompile of all application source modules.

Enable embed commenting

Check this box to optimize automatic comment generation specified by the application templates.

Synchronization

This tab lets you specify how and when control attributes defined in your data dictionary are applied to your application's procedures and controls.

Synchronize Application when opened

Check this box to reapply data dictionary attributes each time your application is opened. Clear the box to apply the attributes only on your explicit command.

Synchronize Window definitions

Check this box to apply data dictionary attributes to WINDOW structures during application-wide synchronization. Clear the box to ignore WINDOW structures.

Synchronize Report definitions

Check this box to apply data dictionary attributes to REPORT structures during application-wide synchronization. Clear the box to ignore REPORT structures.

Update controls for variables

Check this box to apply memory variable control attributes to their associated controls. Clear the box to ignore controls associated with memory variables.

Primary attributes only

Check this box to apply only the primary control attributes. Primary attributes are those attributes set directly in the **Field Properties** dialog. They include Field Name, Characters (length) Screen Picture, Prompt Text, Column Heading, Case (UPR, CAP), Typing Mode (INS, OVER), Flags (IMM, PASSWORD, READONLY), Justification, Initial Value, Help IDs, Messages, Tool Tips, and Validity Checks. Secondary attributes are those attributes set by pressing the **Properties** button on the **Window** and **Report** tabs of the **Field Properties** dialog.

Tip

Check the Primary attributes only box to speed up the synchronization process, especially if you synchronize each time you open the application.

Clear HELP, MSG, TIP if omitted in dictionary

Check this box to override control specific help attributes (set from the **Window Formatter**) with blank help attributes from the data dictionary. Clear the box to retain control specific help attributes despite blank help attributes in the dictionary.

Allow control types to change

Check this box to apply new control types. For example, a SPIN may replace an ENTRY. The synchronizer does not change the number of controls on a window or report; therefore it does not change an OPTION to a LIST or vice versa. However, it does issue a warning when it encounters this situation.

Note:

Changing a control's type can result in "orphaned" embed code. For example a SPIN supports a NewSelection embed point, but an ENTRY does not. Orphaned embed code should be manually moved to an appropriate place.

Allow conversion from list to drop list

Check this box to allow a drop list to replace a list.

Clear all other attributes if omitted in dictionary

Check this box to override all control specific attributes (attributes set from the **Window Formatter**) except HELP, TIP, and MSG with blank attributes from the dictionary. Clear the box to retain control specific attributes despite blank attributes in the dictionary. Clearing the box also enables the **More** button so you can set each attribute individually.

More

Press this button to elect, for each individual attribute, whether to override the control specific (**Window Formatter**) attribute with a blank attribute from the dictionary or whether to keep the attribute despite a blank attribute in the dictionary. Attributes are **Font, Alert, Tally, Cursor, Key, Icon,** and **Colors.**

Dictionary can override size

Check this box to let data dictionary size attributes prevail over **Window Formatter** size attributes. Control sizes can change when the height or width value is default and the control's text changes, or when an explicit height or width value in the dictionary varies from the control specific (**Window Formatter**) height or width values.

Ignore Freeze attribute setting

Check this box to apply data dictionary attributes to controls with the #Freeze attribute. Clear the box to leave frozen controls alone.

If you synchronize a single control, the synchronizer ignores the #Freeze. #Freeze is effective only when you synchronize multiple controls, i.e. an entire application or procedure, a WINDOW structure such as an OPTION or a GROUP, or a REPORT structure such as a DETAIL, BREAK, HEADER, or FOOTER.

Refreeze frozen control after synchronize

Check this box to" refreeze" the control after synchronizing it. Clear the box to "unfreeze" the control after synchronizing.

Update field formatting

Check this box to apply dictionary attributes to LIST FORMAT strings (i.e. justification). In other words, format list box fields according to data dictionary attributes. Clear this box to leave LISTs alone.

Update column headers

Select from the drop-down list to specify when List Box column headers are applied from the data dictionary. Choose from:

Always The Application Generator always applies the dictionary column header, even if it is blank.

If present in dictionary

The Application Generator applies the dictionary column header only if it is non-blank, otherwise, the LISTs column header prevails.

Window and Dictionary

The Application Generator applies the dictionary column header only if *both* are non-blank, otherwise, the LISTs column header prevails. This lets a blank column header on a LIST prevail over a non-blank column header in the data dictionary.

Never The Application Generator never applies the dictionary column header. The LIST's column header always prevails.

Display warning if could not synchronize
Check this box to display a warning dialog if warnings occur during synchronization. Warnings occur when controls change size, when there is a different number of radio buttons in the dictionary than on the window or when an OPTION is replaced with a different control (e.g. a drop-down list).

Add report entry when controls change size
Check this box to generate warnings when controls change size. Clear the box to suppress size change warnings.

Filename for report Specify the file to hold the warning report. Clearing this box suppresses the report.

Embed Editor

This tab lets control how the Application Generator generates the temporary source file for the Embeditor. You can specify the text that delimits the embed points within the temporary source file. By customizing the text, you can make it easy to identify the embed points you want to edit.

Preceding Comment
Specify the text that marks the beginning of each embed point.

Include preceding comment
Check this box to generate a preceding comment. Clear the box to omit the preceding comment.

Prefix Set the text generated before the embed point name.

Suffix Set the text following the embed point name.

Following Comment
Specify the text that marks the end of each embed point.

Include following comment
Check this box to generate a trailing comment. Clear the box to omit any trailing comment.

Prefix Set the text generated before the embed point name.

Suffix Set the text following the embed point name.

Show priority levels

Check this box to show the embedded source priority within the Embeditor. The priority determines the sequence in which the Application Generator places multiple blocks of embedded code within a single embed point.

Edit errors in context

This box controls which edit mode to invoke when you edit embedded source code from the **Make Status** dialog. Check this box to open the Embeditor (equal to **Edit ▸ Source**) to edit embedded source code. Clear the box to open the non-contextual embed editor (equal to **Edit ▸ Embeds**).

Application Explorer

The Application Explorer refers to the main window that displays when an application file is first loaded.

Show complex embed tree
When checked, the filled embeds for a procedure show the expanded embed labels on the right pane. Clear this checkbox to display the embeds without the descriptive labels.

Embed Tree Options

The Embed Tree is the relational list control that displays when selecting the Embeds menu item or button from any selected procedure.

The following settings allow you to fine tune the level of information displayed for all embed points:

Show PROCEDURE Keyword
Check this box to attach the word PROCEDURE to an embed point where appropriate.

Show VIRTUAL Keyword
If a method embed point is defined as a virtual method, check this box to attach the word VIRTUAL to an embed point where appropriate.

Show PROTECTED Keyword
If a method embed point is defined as one that is protected, check this box to attach the word PROTECTED to an embed point where appropriate.

Show Base Class
Check this box to display the base class name related to embed points (WindowManager, FileManager, etc.) where appropriate.

Show Object Description
Check this box to allow a class object's full description to be displayed.

Show Details
Check this box to allow a class object's full details to be displayed.

Color Entries
Check this box to reveal the color dialog described below.

Colors

Embed points can be colored with respect to what section of code it is created in. Modify your colors for DATA and CODE sections, existing VIRTUAL and PROTECTED methods, or new methods that you have created.

Application Tree Dialog - Alphabetic View

In alphabetic view, the **Application Tree** dialog displays your procedures according to the order of the procedure names. A procedure is a collection of instructions--Clarion language statements--which perform a task. The first procedure your application executes is called "Main" by default.

Alphabetic view makes it easier to locate the precise procedure you wish to edit, when your application includes many procedures. There is also a **Find** command on the **Edit** menu, to help you locate the procedure you want when the Procedure tree is *very* long.

The Application Tree shows the procedures you create when you add a menu item, toolbar command, or an embedded source procedure. Each new procedure is marked "To Do." When you "fill in" its functionality, the Application Tree dialog replaces the "To Do" with your description.

Global

Opens the **Global Properties** dialog, which allows you to declare, or edit the declarations of Global data.

The **Global Properties** dialog also allows you to specify your name as the program author, specify that your application store settings such as Window size and position in its own .INI file, and choose the file access modes your application will utilize when it works with its data files.

Properties

Once the procedure appears on the Application Tree, you can define its procedure type by selecting it, then pressing this button. Choose a procedure template from the **Select Procedure Type** dialog.

Once you select a procedure template, you access the other parts of the IDE to determine its functionality through the **Procedure Properties** dialog.

Category View

In this view, the **Application Tree** dialog displays your procedures grouped by the Category you specify on the Procedure Properties dialog. The Procedure templates add a default value corresponding to their types to group procedures made with the same template together. You can create your own categories as you like.

Modified View

In this view, the **Application Tree** dialog displays your procedures sorted by the last modified date and time. The most recent changed procedure appears at the top of the list, while the oldest procedure appears at the bottom. If you ever need to trace when a problem arose, this view can assist you in which procedure could be the problem.

Module View

In module view, the **Application Tree** dialog displays your procedures according to the source code document which they reside in. A procedure is a collection of instructions--Clarion language statements--which perform a task. The first procedure your application executes is called "Main" by default.

The tree controls in this dialog illustrate how the procedures reside in each separate file. Within each, they branch from each other when parent and child procedures reside in the same file.

The Application Tree shows the procedures you create when you add a menu item, toolbar command, or an embedded source procedure. Each new procedure is marked "To Do." When you "fill in" its functionality, the Application Tree dialog replaces the "To Do" with your description.

Procedure View

In this, its default view, the **Application Tree** dialog displays your procedures in logical call tree, nesting each procedure under its calling procedure. A procedure is a collection of instructions--Clarion language statements--which perform a task. The first procedure your application executes is called "Main" by default.

Template Type View

In template type view, the **Application Tree** dialog groups all your procedures according to template type. A procedure is a collection of instructions--Clarion language statements--which perform a task. The first procedure your application executes is called "Main" by default.

BIND Fields and Procedures

The process of BINDing a field, procedure or expression allows the entity be used in an expression string for either the EVALUATE procedure or a VIEW structure's FILTER attribute.

This template interface allows you to specify that a BIND is generated for the fields, procedures, and expressions you designate. The template interface generates the proper syntax for each of these entities.

On the Field tab, you may insert, change or delete a field to be BINDed by selecting the **Developer Defined** tab, and selecting the appropriate button. You will be presented with an ellipsis that will allow you to select any variable defined within your application. The **Template Defined** tab displays any variables that have already been BINDed by other templates. You should always check this tab to verify a list of variables already BINDed.

On the Procedures tab, you may insert, change or delete a procedure to be BINDed by selecting the **Developer Defined** tab, and selecting the appropriate button. You will be presented with a drop list that will allow you to select any procedure name currently defined within your application. You can also add new procedure names not yet defined here. The **Template Defined** tab displays any procedures that have already been BINDed by other templates. You should always check this tab to verify a list of procedures already BINDed.

On the Expression tab, you may insert, change or delete a name to be BINDed to an expression by selecting the appropriate button.

Name Enter a string value to use as the name to be used to identify the BINDed expression.

Expression An expression is a mathematical formula containing any valid combination of variables, functions, operators, and constants. Enter a valid expression to be BINDed. Example: (TODAY() – 1)

Dictionary Viewer

The Dictionary toolbox provides a hierarchical list representing your database. This list shows database files, keys, key components, fields, and relationships in an expanding hierarchical tree. Each list item has a properties dialog for manipulating the item's properties. RIGHT-CLICK on an item to open its properties dialog. CLICK on the plus (+) sign to expand the list; CLICK on the minus (-) sign to contract it.

The Dictionary toolbox can update all the existing items in the active data dictionary except the dictionary properties and any other properties that are already in use within the development environment. The Dictionary toolbox cannot add new items (new files, fields, keys, etc.) and does not call the Database Manager to browse, edit, or convert files.

You can use the Dictionary toolbox to access a dictionary that is already in use within the development environment, with the noted limitations. From the **Dictionary** dialog, choose **Edit ▸ View as Toolbox**, or from the **Application Tree** dialog, choose **Application ▸ View Dictionary**

Edit Control Templates Dialog

This dialog allows you to access the prompts dialog for a control template associated with the current procedure.

This is equivalent to selecting the control template in the Window Formatter, then selecting **Actions**. If, however, additional prompts from field templates, such as the actions for a button when pressed, apply, those are only available through the Window Formatter.

Select a control template from the list, then press the **Properties** button.

Edit Default Procedures Dialog

The template registry allows for multiple starting points for a procedure template. For example, you could have two browses, one of which you wish to use most of the time, and the other, some of the time.

This dialog allows you to set the default. It also allows you to add an alternate procedure.

Add	Allows you to name the alternate, then set its properties in the Procedure Properties dialog.
Properties	Opens the Procedure Properties dialog for the selected procedure.
Delete	Deletes the selected procedure.

Edit Procedure Description

Allows you to enter string descriptions for the procedure. Clarion automatically displays the short description in certain dialogs, allowing you to quickly recognize the file contents. The long text description only appears in this dialog box, and holds up to 1000 characters.

The descriptions are solely for your convenience, and have no effect on the application. They´re useful for situations in which other programmers may pick up your code later, or for when you expect to return to the project after a long period of time since you last looked at it.

Embedded Source Dialog

This dialog allows you to access Embed points from the Procedure Properties dialog.

By adding embedded source code to a procedure, you gain powerful customization capability. You can specify or create code to execute at any defined point in the source code. You can write your own code, or use a code template to write the code for you. The Application Generator adds your code to the code it generates, at precisely the point at which you specify.

This dialog lists all the available embed points, as defined by the procedure template.

Tip

To see your embedded source code in context, use the Embeditor instead (from this window, press the Source button. From the Application Tree, RIGHT-CLICK on the procedure and choose *Source*).

Edit Options

The Edit Menu (and the edit buttons on the right) allow you to manipulate your embedded source code.

Cut (CTRL+X)	Cuts the highlighted embedded source to the clipboard, allowing you to paste it into another embed point in this or any other procedure (including procedures in other applications).
Copy (CTRL+C)	Copies the highlighted embedded source to the clipboard, allowing you to paste it into another embed point in this or any other procedure (including procedures in other applications).
Paste (CTRL+V)	Pastes embedded code from the clipboard into the highlighted embed point.
Insert (INS)	Opens the **Select Embed Type** dialog, which allows you to add handwritten source code, call a procedure, and/or choose a code template.
Properties (ENTER)	Allows you to edit the embedded code. If it is hand written code, then the Text Editor appears. If it's a code template, the prompts dialog for the code template appears.
Delete (DEL)	Allows you to delete embedded code you previously added.

Move Up (CTRL+UP) Moves the embedded code item up above another (modifying the Priority). Each executes in the order they appear at an embed point.

Move Down (CTRL+DOWN)
Moves the embedded code item down below another (modifying the Priority). Each executes in the order they appear at an embed point.

View Options

The View Menu (and toolbar buttons) allow you to adjust the display to show only what you want to see. The View Menu options are:

Expand All Fully expand the embeds list.

Expand Filled Expand only the filled embeds.

Contract All Fully contract the embeds list.

Show Filled Only Show only filled embeds.

Show Priority Labels Show template generated embed point labels so you can precisely interleave your code with template-generated code.

Include Legacy Embeds
Show Clarion 2.x embed points.

Tip

You may set the default for legacy embed points with the Application tab of the Application Options dialog. Choose Setup ▸ Application Options.

Show Window Embeds
Available only when editing embeds for a control, this button allows you to expand the view to show embeds for the window.

Navigation Options

The Navigate Menu (and toolbar buttons) allow you to adjust the display to show only what you want to see. The Navigate Menu options are:

Previous Filled (CTRL+P) Moves the selection to previous filled embed point. It expands the tree as needed.

Next Filled (CTRL+N) Moves the selection to next filled embed point. It expands the tree as needed.

Priority

Priority Sets the **Priority** for the embedded source. The **Priority** of each block within an embed point controls the execution sequence of the code relative to any other code in the same embed point. Lower priority numbers execute before higher priority numbers.

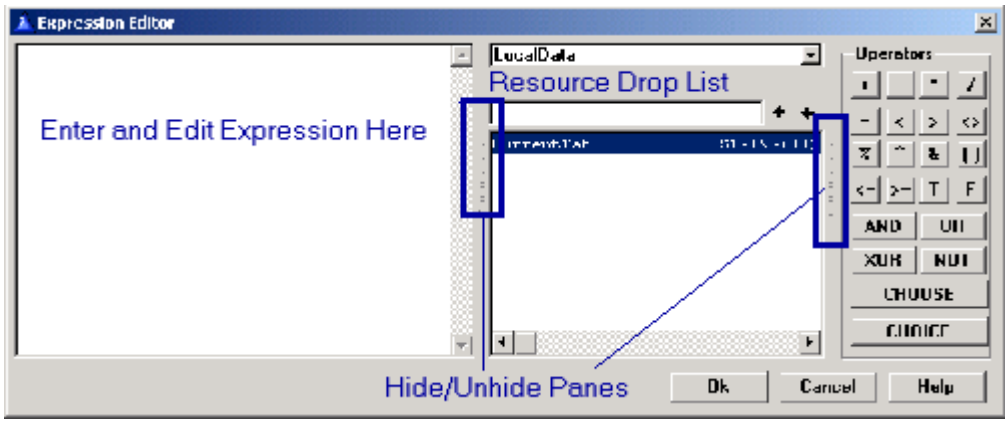
Tip

Moving an embed up or down changes its priority.

Source button

Press this button to call the Embeditor, allowing you to see your embedded source in context.

Expression Editor Dialog



The Expression Editor is a built-in utility that helps you build all types of expressions. (Chapter 10 of the Language Reference provides a detailed overview of Clarion language expressions)

Note:

Through the environment, a button labeled with a boldface E calls the Expression Editor.

Refer to the diagram shown above and the following prompts:

Enter and Edit Expression Here

The left window pane (a text box) is used to display the expression that you are constructing. You can enter an expression manually, or use the helper panes to the right to help you build your expression.

Resource Drop List

Use the drop list shown above to select from a variety of elements to use in your expression. This includes:

Clarion Functions	A list of available built-in Clarion functions
GlobalData	A list of the application's global data elements
ModuleData	A list of the current module's data elements
LocalData	A list of the current procedure's local data elements
Tables	A list of the tables defined in the current procedure
Procedures	A list of procedures called from the current procedure
Window Control	A list of window elements and their field equate labels

Based on the resource selected, additional prompts may appear to help you select additional elements. Locators are provided for all resources selected.

Operators

Provides buttons for inserting logical and bitwise operators into the expression. You can also type them in directly.

Formula Dialog

This dialog lists all formulas already created for a procedure, along with their template classes. It allows you to add or edit formulas.

If any formulas already exist for the procedure, this dialog appears when you push the Formulas button in the Procedure **Properties** dialog.

The list displays the following information:

Formula	Lists all existing formulas within the procedure.
Class	Lists the template class associated with the formula. A formula's class determines when its calculation is performed. Each template has its own set of classes. For example, in the Form Procedure Template there is a class called "After Lookups" which tells the Application Generator to compute the formula after all lookups to secondary files are completed for the procedure.
Description	A short text description of the formula.

The dialog offers the following buttons and their actions:

Select	Loads the currently selected formula into the **Formula Editor** for editing.
New	Loads the Formula Editor, ready to create a new formula.
Delete	Deletes the selected formula.

Formula Editor Dialog

The **Formula Editor** dialog provides access to fields defined in the file schematic, as well as global or local variables, and facilitates creating syntactically correct expressions.

To create an expression, you press buttons to add components to the Statement line. You can also type in your expression, and check the syntax upon completion.

Name	A descriptive label for the formula.
Class	A formula's class determines when its calculation is performed. Each template has its own set of classes. For example, in the Form Procedure Template there is a class called "After Lookups" which tells the Application Generator to compute the formula after all lookups to secondary files are completed for the procedure. Press the ellipsis (...) button next to the field to view the list of available template classes in the **Template Classes** dialog.
Description	A short text description for the formula.
Result	The variable to which the value of the expression is assigned at run time. Press the ellipsis (...) button next to the field to view the **Table Schematic Definition** dialog, in which you can select or define the variable.
Statement	The actual expression under construction.
Check	Tests and validates the expression under construction. A check box appears if the expression is syntactically correct. An "X" appears if not.
Information	Describes the currently selected component in the **Statements** box.
Operators	Provides buttons for inserting logical and bitwise operators into the expression. You can also type them in directly.
Data	Accesses the **File Schematic Definition** dialog where you can select or define a variable or field as an operand within the expression.
Functions	Access a list of built-in Clarion functions in the **Functions** dialog.
User	Accesses user defined functions within the application under development, displaying them in the **User Function** dialog.

Conditionals Accesses the **Conditional Dialog**, which allows you to create a conditional expression.

Lists all of the Clarion built-in functions. Highlight the function you want, then press the **Select** button.

Lists all of the user defined functions in the application. Highlight the function you want, then press the **Select** button.

A conditional field is a computed field with multiple possible expressions. There are two types of conditional fields--IF structures and CASE structures. The assignment statement executed depends on the evaluation of the IF or CASE condition. For example, an IF structure conditional field called Tax could be 0 if Taxable is FALSE, or Price times TaxRate if Taxable is TRUE.

The Formula Editor allows you to create a conditional expression whose result can then be assigned to a variable. Name your conditional formula in the **Formula Editor** dialog, then press the **Conditionals** button to open this dialog.

Each portion of the expression is edited separately. The components appear in the **Structure** list in the lower portion of the dialog box. Select a component, then edit it in the Statement box. You can add and/or nest IF and CASE structures by pressing the **IF THEN** and **CASE OF** buttons.

Statement A currently selected component (displayed in the **Structure** list) of the actual expression under construction.

Information Describes the currently selected component in the **Statements** box.

Check Tests and validates currently selected component of the expression under construction. A check box appears if it is syntactically correct. An "X" appears if not.

Accept Adds the currently selected component of the expression to the **Structure** list.

Structure Lists the components of the expression in a hierarchical list. Each item selected can be edited separately.

Operators Provides buttons for inserting logical and bitwise operators into the expression. You can also type them in directly.

Data Accesses the **Table Schematic Definition** dialog, where you can select or define a variable or field as an operand within the expression.

Functions Access a list of built-in Clarion functions in the **Functions** dialog.

User Accesses user defined functions within the application under development, displaying them in the **User Function** dialog.

IF THEN	Adds and/or nests an IF THEN structure to the expression.
CASE OF	Adds and/or nests a CASE OF structure to the expression.
Delete	Deletes the selected statement from the conditional structure.

Procedure Setup --Upon Entry into the Procedure

This point occurs immediately after the CODE statement, allowing you to initialize values upon entering a procedure.

Before Lookups--Refresh Window ROUTINE, before lookups

This occurs before any lookups to related records, allowing you to prime any key values needed to perform the lookups.

After Lookups--Refresh Window ROUTINE, after lookups

This occurs immediately after looking up related records, allowing you to use values retrieved from related records in your computation.

Procedure Exit--Before Leaving the Procedure

This allows you to assign values before returning to the calling procedure.

Prime Fields--Prime Fields of the Primary File record at beginning of Insert

Available when a Save Button control template is used, this allows you to pre-assign values to fields when inserting a new record.

Before Filter Check--In Validate Record ROUTINE, Before Filter Code

Available when a BrowseBox control template is used, this allows you to create a formula to be used in the filter expression.

Before Range Check--In Validate Record ROUTINE, Before Range Limit Code

Available when a BrowseBox control template is used, this allows you to assign values before range limits checks are made.

Format Browse--Format a variable in the Browse Box

Available when a BrowseBox control template is used, this allows you to compute values to display in the list box.

Before Print Detail--Before Printing Report Detail

Available only when the Report template is used, this allows you to compute values before a sending a detail structure to a report.

Global Data/Local Data/Module Data Dialogs

This dialog allows you to define or edit memory variables. It provides a list of variables for the procedure (Local), module (Module), or program (Global).

When you want to add a variable, you press the **Insert** button, then define the variable in the **New Field Properties** dialog.

Properties	Select a variable from the list, then press this button to edit it in the **Edit Field Properties** dialog.
Insert	Press this button to define a new variable.
Delete	Press this button to delete the currently selected variable.
Up	Press this button to move the currently selected variable up one position in the list. When the Application Generator generates the code defining the data, it will do so in the order they appear in this list.
Down	Press this button to move the currently selected variable down one position in the list.

Global Field Selection Dialog

The Global Field Selection window presents a relation tree of tables defined in the dictionary. Press the Expand (+) button to display columns attached to a given file, and press the **select** button to copy the field name to the appropriate entry.

Global Properties

This dialog specifies application level options for file processing, .INI file support, plus lets you define global variables.

Data
Press the **Data** button to open the **Global Data** dialog, which lets you declare or edit global variables. The ellipsis (...) button next to the **Data** button lets you view or edit the memory variable declarations in TXA format.

You may declare new data items here with normal Clarion language syntax. You do not need to supply the TXA code.

Embeds
Calls the **Embedded Source** dialog for global level embed points including the global map, global data, file open routines, the export file, and the ship list.

Extensions
Calls the **Extensions and Control Templates** dialog to manage global extensions for the application. Extension templates generate code to provide functionality not associated with a specific control.

General

Program Author
Lets you add your own name, which is then added into the .APP file.

Default Icon
Press the ellipsis button to select an icon to use as the program icon. You may also specify a variable name here by prefacing the variable with an exclamation point (!). The templates implement the use of **`SYSTEM{PROP:Icon}`**.

Use Field Description as MSG() when MSG() is Blank
Check this box to tell the Application Generator to use field descriptions from the **Field Properties** dialog as the default MSG attribute.

Generate Template global data and ABC's as EXTERNAL
Adds the EXTERNAL attribute to the global variable declarations generated by the templates, and the DLL attribute to any CLASS declarations generated by the templates. This means your program relies on an external library to allocate memory for these variables and objects, and to export them so your program can access them.

You should add the EXTERNAL and DLL attributes to get the same effect for any global variables or classes you declare. See the *Language Reference* for more information on these attributes.

Note:

If you create a program that consists of more than one AppGen created DLL, you should check the Generate Global Data as EXTERNAL box for all the applications except one. See the *User's Guide--Development and Deployment Strategies*.

External Globals and ABC's Source Module
Specify whether the external library is dynamically or statically linked.

This sets the *flag* parameter of the DLL attribute for template generated class declarations. See the *Language Reference* for more information on the DLL attribute.

Generate Embed Comments
Check this box to generate identifying comments surrounding all embedded source code.

Non-Volatile Storage Settings

The Clarion and ABC Templates non-volatile storage support the use of .INI (standard windows initialization) files or the system registry. These are mediums that store information for an application between sessions.

One use for non-volatile storage is to store the user's preferred window positions for the next session. Another use is to save program configuration settings between sessions. Clarion's procedure templates let you do both automatically when you enable non-volatile storage support.

Location
Choose between *INI file* and *System Registry*

INI File Options

.INI File to use
Specify whether to use the default.INI file name. By default, the application creates the .INI file with the same file name as your application. To use another name select the *Other* choice from the drop down list.

Other File Name
Specify another .INI file name other than the default. You may specify a variable, a full pathname, no path, or a path of (.\) to generate the INI file as shown below.

Other File Name	Resulting INI File Location
!variable	(The contents of the variable specified)
c:\programs\payroll.ini	c:programs\payroll.ini
payroll.ini	(windows system directory)\payroll.ini
.\payroll.ini	(current directory)\payroll.ini

Use App Directory
Check this box to direct your INI file to be located in your application directory (ABC Only).

Registry Options

App Registry Key
Enter the name of the key whose value is to be queried. This may contain a path separated by backslash '\' characters. Example: `SOFTWARE\SoftVelocity\Clarion6`

Registry Root
Use the drop list to select a valid registry root where your registry key will be located.

Preserve
Once you've enabled INI support, the ABC Templates automatically save and restore the values of designated global variables. This provides a simple mechanism for saving and reapplying end user preferences or program configuration options. Press the Data button in the Global Properties dialog to define your global variables, and then press the Preserve button to designate selected variables to automatically save and restore.

In the Preserve dialog window, you have the standard buttons available **Insert**, **Delete**, or examine and modify (**Properties**) the preserved variable. You can also move the variables up and down in the list.

Enable Run-Time Translation
Generates code to translate window text based on the translation strings defined by default in the ABUTIL.TRN file. See *Translator Class* for more information.

Enable Fuzzy Matching
Check this box to generate the necessary objects to support the BrowseFuzzyMatching control template.

Fuzzy Matching Options

Ignore Case
Check this box to enable all fuzzy searches in your application to be case insensitive. This can be overridden individually. See *FuzzyClass- SetOption* method for more information.

Word Only
Check this box to enable all fuzzy searches in your application to search on whole words only. This can be overridden individually. See *FuzzyClass- SetOption* method for more information.

App Settings

Enable Window Frame Dragging
Check this box to disable the "Show Contents While Dragging Window" Windows option. This functionality is known by Microsoft to cause Windows internal messaging system problems which may lead to GPFs. We recommend that the **Enable Window Frame Dragging** remain checked.

Include Default XP Manifest
Check this box to include a default Manifest file in your application. When your program runs in the XP environment, this instructs the XP theme manager to render your controls with the new look and feel of the Windows XP environment. The manifest file is ignored on other Windows platforms (95/98, 2000, NT)

Provide visual indicators on control with focus
This check box refers to certain input controls that will change color when you select them. Check this box to enable the **Set Visual Indicators** button, which allows you to designate the controls and colors to use with this feature.

Field Navigation

Use ENTER Instead Tab
The TAB key is the standard key press in Windows for moving from control to control. If you wish to replace this with the ENTER key (the DOS standard), check this box.

Exclude Controls
Press this button to designate special controls that will NOT use the ENTER key for navigation.

Report Preview Mode (Clarion chain only)
Select the type of Print Preview you wish to use in your applications. Click on **ABC Class** to use the Preview Class used in the ABC template chain, or click **Procedure** to use a default preview procedure found in the Clarion template chain.

If **Procedure** is selected, accept the **Default Report Preview** procedure used by the Clarion template chain, or enter an alternate procedure name (one that you have modified, or obtained from a third-party source.)

Enable Auto Size BrowseBox Columns?
Check this box to enable the auto resize column feature. At runtime, click on the right line of any column and it will automatically resize to fit the data contents. You can disable this feature in the local Browse procedure settings.

Enable List Format Manager?
Check this box to enable the List Format Manager button.

File Control Flags

The **File Control Flags** dialog lets you override some of the settings in your dictionary, as well as define how procedures will access files. You can specify file attributes for all files, or individually.

Generate all file declarations
Generates all file declarations in the dictionary, even if not specified in any procedure's file schematic.

When done with a file (Clarion templates only)
Specifies whether the application automatically closes each file when a procedure is finished.

Tip

One way in which you can design your application to be a "well-behaved" Windows application is not to hog system resources. You can return file handles not in use by checking this box.

Enclose RI code in transaction frame
Enables rollback of data if an update fails.

Tip

If all files in a relation chain use the same file system, and the file system supports transaction framing, and you do not want transaction framing around the RI code, you must clear the check box here, and choose No for *each* file on the Individual File Overrides tab.

Issue template warning if LOGOUT() not allowed
Enables a compile time warning when your data dictionary includes a file driver which does not support the LOGOUT() function.

You should clear this check box for drivers such as dBase III.

Seconds for RECOVER
Specifies the number of seconds to wait before invoking the RECOVER process. This is applicable only to Clarion files.

File Attributes

Threaded
Specifies whether the application generator adds the THREAD attribute to FILE structures. THREAD is needed for MDI browse and form procedures to prevent record buffer conflicts when the end user changes focus from one thread to another.

Use File Setting
Sets the THREAD attribute according to the data dictionary.

All Threaded
Adds the THREAD attribute to each FILE.

None Threaded
Omits the THREAD attribute for each FILE.

Create
Specifies whether your application should allow the creation of a data file should it not exist. Adds the CREATE attribute to the FILE structure.

Use File Setting
Sets the CREATE attribute according to the data dictionary.

Create All
Adds the CREATE attribute to each FILE.

Create None
Omits the CREATE attribute for each FILE.

External
Specifies whether the application generator adds the EXTERNAL attribute to FILE structures. EXTERNAL specifies the memory for the FILE's record buffer is allocated by an external library.

None External
Omits the EXTERNAL attribute from all file declarations.

All External
Adds the EXTERNAL attribute to all file declarations *and* lets you specify the **Declaring Module** and whether **All files are declared in another .APP.**

Note:

When using EXTERNAL to declare a FILE shared by multiple libraries (.LIBs, or .DLLs and .EXE), only one library should define the FILE without the EXTERNAL attribute. This ensures that there is only one record buffer allocated for the FILE and all the libraries and the .EXE will reference the same memory when referring to data elements from that FILE.

Export all File Declarations
Checking this box tells the Application Generator to add the file information to the Export file. This is only available when the project's **Target Type** is a DLL and you specify **None External** (see above)**.**

Declaring Module
The filename (without extension) of the MEMBER module containing the FILE definition without the EXTERNAL attribute. If the FILE is defined in a PROGRAM module, leave this field blank.

All files are declared in another .APP
Checking this box tells the Application Generator that the files are declared in another APP (rather than hand code). Application Generator adds the EXTERNAL flag on the File:Open flag (that controls CheckOpen), ensuring that files are opened and closed at the right time, thereby preserving the integrity of the file data buffers.

File Open Mode Specifies how your application opens files.

Open
Opens files as Read/Write(primary user) + Deny Write(all other users).

Share
Opens files as Read/Write(primary user) + Deny None(all other users).

Other
Specify a custom combination of primary user + other user access.

User Access
Choose from *Read Only, Write Only,* or *Read and Write.*

Other Access
Choose from Deny None, Deny All, Deny Read, Deny Write, or Any Access (FCB compatibility mode).

Individual File Overrides

Select the **Individual File Overrides** tab to override data dictionary settings for individual files in the data dictionary. Highlight the file whose attributes you want to change, then press the **Properties** button.

The prompts on this tab mirror those on the **File Control Flags** tab, and they behave exactly the same way, with these exceptions.

- The settings here apply only to the single file selected.
- Each drop-down list provides an additional choice: *Use Default. Use Default* sets the attribute according to the **File Control Flags** tab.

File Manager Options for *Filename*
Press this button to display the File Manager Options dialog. You may derive your own new methods or properties for the File Manager Class, or you may specify your own third party class to override the default File Manager Class.

Relation Manager Options for *Filename*
Press this button to display the Relation Manager Options dialog. You may derive your own new methods or properties for the Relation Manager Class, or you may specify your own third party class to override the default Relation Manager Class.

File Declaration Mode
Select from the drop down list one of the four following options. The table in which you are overriding properties for will be generated as a FILE, QUEUE, or GROUP.

Use User Options
The table is generated based on the user options defined in the dictionary. The user options can be set to DATA,QUEUE or DATA,GROUP. This will cause the table to be generated as a QUEUE or GROUP respectively. No user options will cause the table to be generated as a FILE. The **File Access** prompts for **User Options** are the same as the prompts on the **File Control** tab.

As FILE
The table will be generated as a FILE structure. The File Access prompts for User Options are the same as the prompts on the File Control tab.

As GROUP
The table will be generated as a GROUP structure with the possibility of the attributes below.

THREAD
Check this box to add the THREAD attribute to the GROUP structure.

BINDABLE
Check this box to add the BINDABLE attribute to the GROUP structure.

NAME
Specifies the value of the NAME attribute added to the GROUP structure.

OVER
Specifies the value of the OVER attribute added to the GROUP structure.

TYPE
Check this box to add the TYPE attribute to the GROUP structure.

As QUEUE
The table will be generated as a QUEUE structure with the possibility of the attributes below.

THREAD
Check this box to add the THREAD attribute to the GROUP structure.

BINDABLE
Check this box to add the BINDABLE attribute to the GROUP structure.

NAME
Specifies the value of the NAME attribute added to the GROUP structure.

OVER
Specifies the value of the OVER attribute added to the GROUP structure.

TYPE
Check this box to add the TYPE attribute to the GROUP structure.

File Open Mode
Specifies how your application shares files among concurrent users. See the *Language Reference* for more information.

Open Opens file as:
Read/Write (primary user) +
Deny Write (all other users).

Share Opens files as:
Read/Write (primary user) +
Deny None (all other users).

Other Specify a custom combination of primary user + other user access.

User Access
Choose from Read Only, Write Only, or Read and Write.

Other Access
Choose from *Deny None*, *Deny All*, *Deny Read*, *Deny Write*, or *Any Access* (FCB Compatibility mode).

Defer Opening File
Specifies when your application opens related files. Select Yes to delay opening the file until it is accessed. Delaying the open can improve performance when accessing only one of a series of related files. Select No to open the file immediately whenever a related file is opened. See: *File Manager Class – Lazy Open* and *Use File* in the ABC Library Reference for more information.

External Module Options

Select the **External Module Options** tab to set options associated with your application's external modules. This tab is only available when your application contains an external module (LIB or DLL). Select the external module whose attributes you want to change, then press the **Properties** button.

Standard Clarion 6 LIB/DLL
Check this box if the LIB or DLL is produced by the ABC Templates or a similar coding scheme. Checking the box generates code to initialize and shut down global objects used by the LIB or DLL. If it is a hand-coded LIB or DLL you should probably clear this box.

Global Objects

This tab lets you specify the default object names the global objects used in an application. You can also specify the default classes to be used for the global objects.

Don't generate globals
Check this box to not generate any global data for the application. This includes FILE and object declarations. It allows you to generate one DLL that is shared by all applications.

Error Manager
Press this button to display the Error Manager classes dialog. You may derive your own new methods or properties for the Error Manager Class, or you may specify your own third party class to override the default Error Manager Class.

INI File Manager
Press this button to display the INI File Manager classes dialog. You may derive your own new methods or properties for the INI File Manager Class, or you may specify your own third party class to override the default INI File Manager Class.

Run-time Translator
Press this button to display the Run-time Translator classes dialog. You may derive your own new methods or properties for the Run-time Translator Class, or you may specify your own third party class to override the default Run-time Translator Class. (Checking the Run-time translation check box on the General Tab enables this button.)

Fuzzy Matcher
Press this button to display the Fuzzy Matcher classes dialog. You may derive your own new methods or properties for the FuzzyClass, or you may specify your own third party class to override the default Fuzzy Matcher class. (Checking the Enable Fuzzy Matching check box on the General Tab enables this button.)

Classes

Enable the use of ABC Classes (Clarion Family Only)
Check this box to allow the ABC Classes to be used with the Clarion template family. This is required for certain templates that are used in both families and references ABC Classes. The default is on (checked).

Refresh Application Builder Class Information
Press this button if you have changed the contents of or added an include file (.INC) to the \LIBSRC directory. Typically, this is needed when you install third party products that use ABC compliant classes, although you may create your own ABC compliant classes too. The ABC Templates use information gleaned from the header files for generating embed points, loading the Application Builder Class Viewer, application conversion, etc.

Application Builder Class Viewer
Press this button to display classes, properties, and methods used by the ABC Templates, and the relationships between parent and derived (child) classes. This utility can help you analyze and understand the classes that the ABC Templates use.

Task Grouping Buttons
Each task-grouping button identifies tasks or types of tasks the ABC Templates accomplish. Each button lets you specify the class or classes the ABC Templates use to accomplish the tasks named by the button's text. Following are the ABC Template tasks and their associated default classes.

General Tasks

This window allows you to specify the name of the Class that handles certain tasks in your application. You may specify alternate classes by typing the class name in the corresponding entry field. The class you name must be an ABC compliant class.

The Configure buttons allow you to set options that affect the behavior of the named class.

Configure WindowManager

Reset on gain focus
Check this box to make the WindowManager unconditionally reset whenever the window receives focus. Clear the box to allow a conditional reset (reset only if circumstances demand, for example, when the end user invokes a new BrowseBox sort order or invokes a BrowseBox locator).

Auto Tool Bar
Check this box to make the WindowManager try to set the appropriate ToolbarTarget whenever the end user selects a new TAB control. Clear the box to manually set the ToolbarTarget or use the current ToolbarTarget.

Configure ErrorManager

Default Error Category
Specify a default error category for errors that do not have their own category. The default is ABC.

Store Error History
Check this box to save error history to be viewed at a later time during the session.

Limit Stored History
Check this box to limit the number of history items the history queue can hold at a single instance.

History Threshold Limit
Specifies the maximum number of items to hold in history. This is available if **Limit Stored History** is checked.

View Trigger level
Select an error severity level from the drop-down list. When the error level occurs the action is recorded to history. The available error levels are Level:Benign, Level:Cancel, Level:Notify, Level:User, Level:Program, and Level:Fatal.

Configure Resizer

Automatically find parent controls
Check this box to make each Resizer object set parent/child relationships among window controls. Clearing the box makes the WINDOW the parent of all its controls. Setting parent/child relationships lets any special scaling cascade from parent to child.

Optimize Moves
Check this box to move all controls at once during the resize operation, producing a snappier resize and avoiding bugs on some windows.

Optimize Redraws
Check this box to make controls transparent (TRN attribute) during the resize operation, producing a smoother redraw and avoiding bugs on some windows.

Configure Run-time Translator

Extract Filename
Specify a filename to receive a list of all runtime text that may require translation for multi-language applications.

This is only available if you have checked Enable Run-Time Translation in the Application's Global Properties.

Configure Calendar

Change Default Color
Check this box to allow you to override the default colors of the selected CalendarClass. You can change the color for Sunday, Saturday, Holiday, and all Other dates.

Action for Close Button
Each Calendar Class has a default Select and a default Close button. If you would like both buttons to return the current date selected, choose *Select and Close* from the drop list. If you only want the Select button to return the date, choose *Cancel* from the drop list.

Browser Tasks

This window allows you to specify the names of the Classes which handles certain tasks in your application's browsers. You may specify alternate classes by typing the class name in the corresponding entry field, or select a class name in the drop list provided. The class you name must be an ABC compliant class.

The **Configure** button allows you to set options that affect the behavior of the Browser class. Additional browser related classes are configured by selecting the appropriate tab control at the bottom of this dialog window. The classes that you can configure at the time of this release are EIP (Edit-In-Place), QBE (Query-By Example), Step and Locator Managers, and Others (which include Fuzzy, Grid, and Sidebar Managers).

The EIP Configuration has a special Template Interface prompt which controls the level of configuration that you can apply to your target browse. Select *Original* to control EIP behavior with limited template prompt control. Select *Detailed* for more specific control for each column that has EIP enabled.

For more information concerning these classes, refer to the ABC Library Reference.

Configure Browser

Active Invisible
Check this box to fill the browse queue even when the browse LIST is "invisible" because it is on a non-selected TAB or is otherwise hidden. This improves performance for procedures with invisible browse lists; however, buffer contents for the invisible browse list should not be relied upon. Clear the box to suppress the refill when the listbox is hidden.

Allow Unfilled
Check this box to allow a partially filled LIST when the result set "ends" in mid-list. This improves (SQL) performance by suppressing additional reads needed to fill the list. Clear the box to always display a "full" list.

Retain Row
Check this box to maintain the highlight bar in the same list row following a change in sort order, an update, or other browse refresh action. Clear the box to allow the highlight bar to move.

Process & Report Tasks

This window allows you to specify the name of the Class which handles certain tasks in your application. You may specify alternate classes by typing the class name in the corresponding entry field. The class you name must be an ABC compliant class.

Ascii Viewer Tasks

This window allows you to specify the name of the Class which handles certain tasks in your application. You may specify alternate classes by typing the class name in the corresponding entry field. The class you name must be an ABC compliant class.

Toolbar Manager Tasks

This window allows you to specify the name of the Class which handles certain tasks in your application. You may specify alternate classes by typing the class name in the corresponding entry field. The class you name must be an ABC compliant class.

ABC Library Files
This dialog allows the ABC libraries to be linked in to the application, linked in as an external DLL, or linked in as an external library. If you choose external DLL or external library, the external library base name must be specified.

Clarion Version Information

This tab displays the latest version and family information for the active template set. This information is useful for your own documentation and SoftVelocity Technical Support for reporting purposes.

Global Classes Window

Many of the ABC Procedure, Control and Extension templates provide a Classes tab or dialog. These local Classes tabs let you control the classes (and objects) your procedure uses to accomplish the template's task—that is, they override the global class settings specified here in the **Global Properties** dialog. Deriving your own class can give you very fine control over the procedure when the standard Application Builder Class is not precisely what you need.

Object Name

Set the object's label for the template-generated code.

Use Default Application Builder Class?

Check this box to use the default Application Builder Class. Clear this box to use a class other than the default, and to enable the following prompts.

Use Application Builder Class?

Check this box to select a class from the **Base Class** drop-down list. The list includes all classes with the LINK attribute in \LIBSRC*.INC files. Clear this box to specify a class declared elsewhere.

Base Class

If you checked the **Use Application Builder Class?** box, select a class from the drop-down list. If you cleared the **Use Application Builder Class?** box, type the class label here, and type the name of the source file that contains the class declaration in the **Include File** entry box.

Include File

If you cleared the **Use Application Builder Class?** box, type the class label in the **Base Class** entry box, and type the name of the source file that contains the class declaration here.

Derive?

Check this box to derive a class based on the parent class specified above and to enable the **New Class Methods** and **New Class Properties** buttons to define any *new* properties and methods for the derived class.

This prompt is primarily to allow you to define *new* properties and methods in a derived class. To override *existing* methods, simply embed code in the corresponding method embed points.

Using **Derive?**, **New Class Methods** and **New Class Properties** makes the template generate code similar to the following:

```
MyProcess CLASS(Process)   !derive a class from the parent class
 NewMethod   PROCEDURE      !prototype new class method
 NewProperty BYTE           !declare new class property
           END
```

Tip

The template automatically derives from the parent class if you embed code into any of the derived method embed points, regardless of the status of this check box.

New Class Methods

Press this button to specify the *new* method prototypes to generate into the derived CLASS structure. This opens the **New Class Methods** dialog (see *New Class Methods*).

New Class Properties

Press this button to specify the new property declarations to generate into the derived CLASS structure. This opens the **New Class Properties** dialog (see *New Class Properties*).

Refresh Application Builder Class Information

Press this button if you have changed the contents of an include file (.INC) or added an include file to the \LIBSRC directory. Typically, this is needed when you install third party products that use ABC compliant classes, although you may create your own ABC compliant classes too. See *ABC Compliant Classes* for more information. The ABC Templates use information gleaned from the header files for generating embed points, loading the Application Builder Class Viewer, application conversion, etc.

Application Builder Class Viewer

Press this button to display classes, properties, and methods used by the ABC Templates, and the relationships between parent and derived (child) classes. This utility can help you analyze and understand the classes that the ABC Templates use.

Composite Class

Press these buttons to open a Classes dialog for each class used by the parent class specified above. For example, the WindowManager uses a Toolbar class, so the WindowManager's Classes dialog contains a Toolbar Class button to open a Classes dialog for its Toolbar Class.

List Format Manager Configuration

The List Format Manager is designed to give you the ultimate control of your Browse Box formatting (display options) to your users at runtime. You are essentially creating a table (file) that stores an unlimited amount of list box formatting strings, which are read and applied to your target browse box at runtime. Other stored formats are available via a popup menu, as well as a Format Editor to define new ones. These options are set on the procedure level, and can be accessed here.

Note:

When using the List box Format Manager during development, removal of a column from the list box may result in errors if the removed column is still referenced in a previous saved list box format. To avoid this, modify the saved list box format prior to removing any referenced columns from the list box.

The list format file can be stored in the active data dictionary, or defined within the specific application.

The following prompts are provided:

UserID Field
Enter a number to assign to all list formats used by this application. You can assign a different number to another application, which will only access special formats for that application.

Table Origin
Your table that stores the list format information can be created by your Application, or can be a table that is defined in your Dictionary.

We recommend that you allow the Format Manager to create a file for you (Application), and later import it into an existing dictionary. This will allow you to share formats between different applications that use the same data dictionary.

Table Configuration

If your runtime format file is to be defined in the application (instead of dictionary storage), press this button to access the Table Configuration dialog.

Dictionary Table
If you have selected Dictionary as the Table Origin choice, press the ellipsis to select the appropriate table name from your active dictionary.

The default file format is a TOPSPEED file, and is shown as follows:

```
LFM_CFile FILE,PRE(CFG),CREATE,DRIVER('TopSpeed'),THREAD,NAME('Formats.FDB')
key_Main  KEY(+CFG:ProcId,+CFG:UserId,+CFG:CtrlId,+CFG:FormatId),OPT,NOCASE
Record    RECORD,PRE()
ProcId     SHORT                            ! Procedure identifier
UserId     SHORT                            ! User identifier
CtrlId     SHORT                            ! Control identifier
FormatId   SHORT                            ! Format identifier
FormatName STRING(30)                       ! Format name
Flag       BYTE                             ! Default/current flag
Format     STRING(5120)                     ! Format buffer
VarLine    STRING(2048)                     ! Variable buffer
          END
         END
```

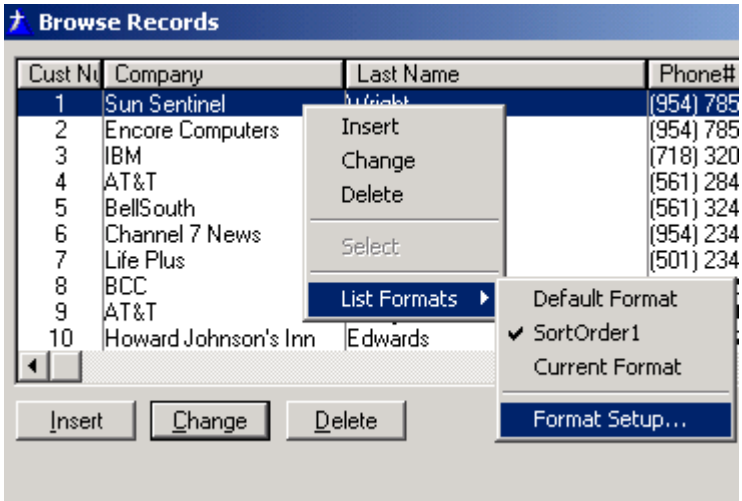
The file driver that you use from the dictionary may be modified, provided that the field and key labels are not modified.

Some elements of the format file can be modified as shown in the Table Information dialog.

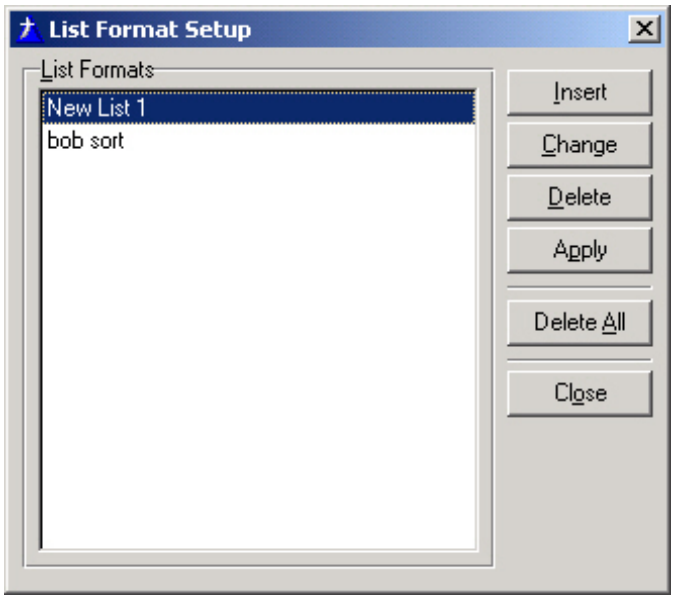
List Format Manager - Procedure Level

The List Format Manager dialog allows you to set the precise level of control for the runtime Format Manager. Special formatting of the list box can be controlled behind the scenes by program logic, or you can open up a special popup design interface that allows the users to create and modify unlimited list box format appearances.

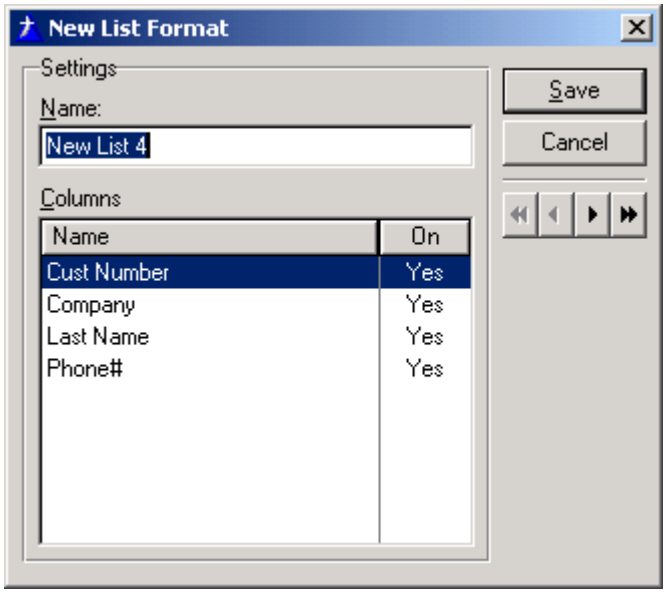
The following images show an example run time Format Manager interface:



1. From the browse box, the user can right-click, and select a different format from the popup menu, or add a new format by selecting the List Format menu item.



2. The list box above shows the Format Editor's active list formats, and allows the user to add, change or delete them as needed.



3. The user has the ability in the above Format Editor window to designate what fields (columns) to use in the new format, and move the field values up or down in order.

Note: If you change the order that the columns are presented in the list box, each column MUST HAVE a specific column field number defined in the list box formatter (do not use the Auto option).

The following prompts are presented:

Popup Mode
Choose Enable from the drop list to allow the user to access the Format Editor interface, or disable to remove this menu item from the popup.

Format Editor Interface:

Items Check Type
Select **Icons** to allow a check box icon to appear to the right of the column selection interface. Selecting **Text** will give you a "Yes" displayed when a column is included in the format string.

Items Sort By
Select **Alpha** from the drop list to display all list formats in alphabetical order. Select **Code** to sort the list formats "as is", in the order that they are entered.

Identificator Tip
Select **Enable** to allow tool tips to be displayed in the Format Editor.

Save Location
Select Enable to allow the user to save the window position of the Format Editor design interface

Different Format for each Sort Order?
Check this box if you would like to save a different list format for each sort order defined. For example, a sort by account number may display a list format with the account number first, a sort by name may display a list format with the name first, etc.

Save Format on Session?
Check this box to allow users to save their format changes when the window is closed.

List Format Table Configuration

The List Format Manager's Table Configuration dialog is used to define your table (file) parameters that will be used to create the list box format configuration file. This file is valid only for this application, but can later be imported and stored in your data dictionary. Note: If you wish to create a runtime format file that is used in multiple applications, define and reference it through a data dictionary.

The following prompts are presented to help you define your format file:

File Group:

Name
Enter a valid file label name to use as the format file. The default name is LFM_CFile

Prefix
Enter a prefix to use for the format file. The default prefix is *CFG*

File Attributes Group:

Check the **OEM**,**Thread**, and **Encrypt** boxes if you wish to apply these attributes to your format file. We recommend that you check the Thread box, to allow the List Format to be used across different thread instances.

Password
Enter an optional password if you wish to encrypt this file for any reason.

Record Group:
The record group allows you to tailor the format file size. For example, if you are using very large list boxes, you may need to increase the format size.

Name Size refers to the size of the Format name that will display in the popup menu and Format Editor. *Format Size* holds the list box FORMAT string, and *Variable Size* holds your formatting options. All are STRING data types, and should be sized accordingly.

File Name Group:

Path
The default path of the format file is the program's directory (folder). If you need to store this file in another location, enter the appropriate path.

Note:

Make sure that your applications that share a common directory specify a different name or path for each application. This can avoid possible conflicts with stored format strings.

Name
The default name of the format configuration file is *FORMATS.FDB*. If you need to change this for any reason, choose **Other** from the drop down list, and enter the new name in the **Use Name** prompt provided.

Module Properties Dialog

This dialog allows you to specify settings for an individual source code document file. You must first view the Application Tree in module view to access this dialog. To do so, you choose **View ▸ Module** View from the IDE menu. Then press the **Properties** button to open this dialog.

Name	Allows you to specify the file name for the module.
Description	Allows you to add a short description, which appears in the Application Tree when in Module View.
Type	Allows you to choose from the **Select Module Type** dialog.
Allow Repopulate	Specifies the Application Generator may move procedures from this and other modules.
Map Include File	Allows you to specify a source code file to include in the data declarations section of the module.

Select Destination Module Dialog

This dialog allows you to manually move a procedure from one module (source code document) to another. Select a module from the list, and then press the **Select** button to move it.

Select Items to Export as Text Dialog

This dialog allows you to choose a procedure from the current .APP file, then export it to a .TXA file for incorporation into another .APP file.

You can select an item by DOUBLE-CLICKING on it. A check mark appears to indicate the item is selected. Select additional items by DOUBLE-CLICKING. De-select an item by DOUBLE-CLICKING a previously selected item.

When your selections are made, press the **Select** button to export them.

Select Items to Import Dialog

This dialog allows you to choose a procedure from another .APP file to import into your current application.

You can select an item by DOUBLE-CLICKING on it. A check mark appears to indicate the item is selected. Select additional items by DOUBLE-CLICKING. De-select an item by DOUBLE-CLICKING a previously selected item.

When your selections are made, press the **Select** button to import them.

Select New Dictionary Dialog

This dialog allows you to change the data dictionary for the current application.

This can introduce problems, since you must ensure that all files and fields referenced in any procedures are present in the new dictionary. Additionally, changing pre-formatting for controls, file relationships, and file driver types can introduce more problems.

Therefore, the dialog box contains a warning that there are no guarantees that changing a dictionary file will work for every application.

To change the dictionary, type a new dictionary file name in the **New Dictionary** box, or press the ellipsis button (...), then select a file from the Open File dialog.

Tip

To avoid these potential problems, create a new application which uses the new dictionary, then import procedures from the original .APP file.

Select Parent Instance Dialog

If a control template needs to attach itself to another, and there is more than one "candidate" to attach to, this dialog appears and allows you to specify which "candidate."

Select the control template you wish to associate, and then press the **Select** button.

Set Visual Indicators

This dialog window allows you to specify the properties to apply to targeted controls when they are selected. Choose from the following options:

Include These Controls

Check the appropriate box to allow the Visual Indicator properties to be applied to specific control types. Choose from TEXT, SPIN, RADIO, CHECK, and DROP LIST control types. When visual indicators are active, ENTRY controls are always enabled.

Control Color Indicator Tab

Change Color
Check this box to allow a control to change its color and text properties when it receives input focus (e.g., is selected).

Background Color
Press the ellipsis button to select the color to use as the background of a control when t is selected.

Bold Text
Check this box to allow any text displayed and entered to be shown in Bold text. When you move to the next control, the text returns to normal.

Exclude DropList
Check this box if you do not wish to apply the selected color entered above to your Drop List controls, but still wish the Bold Text option to be applied. If you want to disable the Visual Indicator feature entirely for all Drop List controls, check the appropriate box located in the **Include These Controls** section above.

REQUIRED fields - Override settings?
Check this box if you would like any fields with the REQ attribute set to be handled with a different color than other selected fields.

Required Fields – Background Color
Press the ellpsis button to select the color to use as the background of the selected required entry or text control.

Box Indicator Tab Control

Display Box?
Check this box to allow a box (with color options) to appear around any control with selected focus. When enabled, the color, border, and size options that follow are enabled.

Fill Color
Press the ellipsis button to select the color to use as the box fill used with the selected control.

Border Color
Press the ellipsis button to select the color to use as the box border used with the selected control.

Size
Set the thickness of the box's border by entering a point value in the spin control. The default is 2 points.

Override settings?
Check this box if you would like any fields with the REQ attribute set to be handled with a different color than other selected fields.

REQUIRED Fill Color
Press the ellipsis button to select the color to use as the box fill used with the selected control that is required.

REQUIRED Border Color
Press the ellipsis button to select the color to use as the box border used with the selected control that is required.

Visual Indicator Tab Control

Display Indicator
Selected fields can also have a special indicator that will be placed immediately to the left side of the selected control type. Check this box to activate a visual indicator.

Indicator Color
Press the ellipsis button to select the color to use with the visual indicator.

Indicator Character
Enter a special character string that you wish to use as the visual indicator (Example: >>)

Offset
Enter the distance in dialog units that separate the visual indicator from the actual selected control.

There is also an **About** tab control located on this dialog window that provides additional information about this feature.

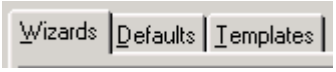
Exclude Controls

The Exclude Controls dialog window is used to Add, Change, or Delete a list of controls (via their Field Equate Label) that will not be affected by the Override Keystroke – Use ENTER instead of TAB feature.

> The Exclude Control prompt requires the Field Equate Label of the fields that you wish to exclude from the Override Keystroke feature. For example, if you want the ENTER key to press the OK and Cancel buttons on all forms (instead of moving past them), enter ?OK and ?Cancel respectively.

Text fields are not affected by the Override Keystroke feature.

Select Procedure Template or Defaults



Wizards

Template Wizards are the most powerful design tool within the Application Generator. Here, you can select a base procedure (Browse, Form, Report, etc.) or a whole application, and then fine tune the wizard prompts and options to produce a near perfect application or procedure that fits your specifications.

See Template Wizards for more detailed information.

Defaults

The Defaults tab allows you to select from a wide variety of pre-defined structures and functionality. Based on the type of default you select, the template procedure associated with it is automatically attached upon your selection.

Templates

This dialog lets you choose a procedure template, adding functionality to any new or "To Do" procedure in the **Application Tree**.

CLICK on a procedure template from the list, then press the **Select** button. Once you select a procedure type, you can customize it using its **Procedure Properties** dialog.

If you add third party, or your own customized templates to the Template Registry, they appear in the list. The following lists the procedure templates that ship with Clarion:

Browse	Browse fields in a page-loaded list box.
External	A procedure contained in an external library (*.LIB only) or object file
Form	View/edit a single record from file
ViewOnlyForm	View a record from a file
Frame	Multiple Document Interface (MDI) main menu
Menu	Single Document Interface (SDI) menu

Process	Sequential record (batch) processor
Report	Generic reporting procedure
Source	Source procedure--add hand-coded source to your application
Splash	Display a splash screen
Viewer	View an ASCII text file
Window	Generic window handler

Select Embed Type

This dialog lets you specify what to embed at this point. CLICK on the item to embed, then press the **Select** button.

The choices are:

Call a procedure Provides a dialog that lets you specify a procedure to call.

SOURCE Calls the Text Editor, allowing you to hand code the embedded source.

Code Templates The templates listed here depend on the templates you have registered. Code templates generate executable code. Generally, each Code template has one well-defined task. For example, the Initiate Thread Code template simply starts a new execution thread, and no more. Typically, the Code template provides a dialog box with options and instructions.

The following Code Templates are included with Clarion for Windows:

CallABCMethod

Call Procedure As Lookup
This code template lets you call a procedure, usually a Browse, with a request to make a selection.

Close Current Window This code template simply posts an EVENT:CloseWindow, which tells the currently active window to close.

Control Value Validation
This code template validates the value of an entry control (ENTRY, LIST, COMBO, or SPIN). You can add this code template to a field event on a control; at the Accepted or Selected embed point.

DisplayPopupMenu

Initiate Thread This code template initiates an execution thread when opening an MDI window.

Lookup Up Non-Related Record
This code template is used to perform a lookup of a value based on a relationship not defined in the Data Dictionary (ad hoc relations). You can add this code template to the Lookup Up Related Records embed point.

ResizeSetStrategy

SelectToolbarTarget This Code template provides an easy way to control which Browsebox is tied to the Toolbar navigation buttons (see **FrameBrowseControl**).

SetABCProperty

SetProperty This Code template provides an easy way to set a runtime property of any control on a window.

Select Utility Dialog

A Utility template lets you produce output from your application. These templates can provide extensible supplemental utilities for such things as wizards, program documentation, or a tree diagram of procedure calls.

Highlight the desired utility template, then press the **Select** button.

Clarion provides ***WIZARDS***--powerful utility templates that let you create a Browse, Form, or Report procedure by answering a few quick questions. You can even use a wizard to create an entire Application from an existing dictionary!

Options you specify in advance in the Data Dictionary provide additional control over the procedures the wizards create. See **Using Wizard Options** for more information.

Application Wizard Creates a complete application from an existing dictionary.

Browse Wizard Creates a multi-keyed browse procedure from an existing dictionary file.

Dictionary Print Wizard
Prints information from your data dictionary, from full detail to high level summary.

Form Wizard Creates an update procedure from an existing dictionary file.

Quick Start Wizard Creates a Data Dictionary and an application based on the dictionary.

Report Wizard Creates multi-keyed report procedures from an existing dictionary file.

Select Extension Template

Extension templates add functionality to procedures, but are not bound to a particular control or embed point. Each Extension template has one well-defined task. For example, the Date Time Display lets you display the date and a running clock.

If you add third party, or your own customized templates to the Template Registry, they appear in the list.

From a **Procedure Properties** dialog, add an Extension template by pressing the **Extensions** button. CLICK on an extension template from the list, then press the **Select** button.

Clarion provides the following Extension templates:

ASCII View in List Box
This extension provides a LIST control to alternate its display between a selected file and some other data that you specify.

cwRTFGloball This global extension enables the use of an RTF control in the application. It includes all necessary EQUATEs and Classes.

Date Time Display This extension adds a "live" date and/or time (updated every second) display to the procedure

DBAuditing The DbAuditing template is a global extension used to track updates to specific data tables. This extension is added to the global properties of an application.

FileUserTags This global template extension identifies certain tables and columns that possess a particular **User Options** tag defined in the dictionary.

Record Validation This extension enables enforcement of dictionary-defined field value validation

RunCommandLineProc
The RunCommandLineProc template provides a simple way to check for command line parameters when the program is run. Several command line parameters may be defined and sorted in the order they should be checked at runtime.

FormVcrControls This Extension template adds functionality to a Form procedure by enabling navigation and field history with the *FrameBrowseControl* VCR buttons. See *FrameBrowseControl* for more information on these buttons and their operation.

ViewFormActions This Extension template adds functionality to a Form procedure by enabling it to open in View Only mode. You specify what mode to set for each control.

WindowResize This Extension template lets the end user resize windows that have traditionally been fixed in size due to the controls they contain (List boxes, entry controls, buttons, etc.).

Wise-Generate Wise Installation Script
This Extension template automatically generates a Wise Installation script for your application.

Procedure Properties

These dialogs--each is customized according to the procedure template--contain entry boxes in which you can add a text description for the procedure, or specify its source code module, plus command buttons which lead to the dialogs which allow you to customize the procedure.

Each procedure has its own custom help page which you can access by pressing the **Help** button on the **Procedure Properties** dialog. If you are using a third party template, or a template which you wrote yourself, this help topic will appear.

Therefore, this help page only describes the essential elements of the **Procedure Properties** dialog; the controls which each procedure template builds upon.

Clarion's Template language allows the template writer to add controls to the **Procedure Properties** dialog. These controls vary from template to template. Since each template performs a different task, the template writer provides whatever controls and options are necessary to gather input from you, the developer. Most of your input is stored in template variables (Template Symbols). When generating code, the Application Generator processes the template language code, and fills in the Template Symbols with the options you specify. As it does so, it generates your application's source code.

A **Procedure Properties** dialog could have, for example, a checkbox to specify that an MDI window should save its position in the .INI file between sessions. Each template adds controls such as these to the **Procedure Properties** dialog, to gather choices from you. At code generation time, the Application Generator evaluates a symbol which stored your choice as to whether you wanted to save the MDI window position. If the checkbox was marked "yes," the Application Generator processes the template code containing the executable code to support saving the window position, and writes it to the generated source code file.

Description

A short text description for the procedure, which appears next to the procedure name in the **Application Tree** dialog.

Press the ellipsis (...) button to edit a longer (up to 1000 characters) description.

Prototype

Allows you to optionally type a custom procedure prototype which the Application Generator places in the MAP section.

Module Name

The source code file to hold the code for the procedure. Select from the dropdown list. By default, the Application Generator names modules by taking the first five characters of the .APP file name, then adding a three digit number for each module.

Parameters Allows you to specify parameter names (an optional list of variables separated by commas) for your procedure, which you can pass to it from a calling procedure. You must specify the functionality for the parameters in embedded source code.

Return Value Lets you specify the variables receiving return values from functions (functions return values, procedures do not).

Save and Restore Window Location
Check this box to make this procedure save and restore its window location using an .INI file.

Tables Accesses the **Table Schematic Definition** dialog. You can define the procedure's access to variables or other files through the dialog.

Window Calls the Window Formatter, to visually design the window.

The ellipsis (...) button next to the **Window** button allows you to edit the WINDOW or APPLICATION structure at the source code level. Clarion allows you to easily switch back and forth between editing the window graphically, and editing the source code that describes it.

Tip

Take care when hand-editing code for any WINDOW which contains a control template. The Application Generator stores Template Language attributes which cannot be edited by hand (#ORIG, #LINK, etc.).

Report Press this button to call the Report Formatter to visually design the window.

The ellipsis (...) button next to the **Report** button allows you to edit the REPORT structure at the source code level. Clarion allows you to easily switch back and forth between editing the report graphically, and editing the source code that describes it.

Data Adds or edits local variables. Press this button and fill in the **Local Data** dialog. Any variables defined are local to the procedure. Define global variables by pressing the **Global** button in the **Application Tree** dialog.

The ellipsis (...) button next to the **Data** button allows you to view the memory variable declarations at the source code level.

Procedures

Calls procedure made in hand-coded, embedded source.

Press this button to access the Called Procedures dialog. To add a procedure, press the **Insert** button, and type a procedure name in the next dialog.

If procedure calls already exist, the names appear in the **Called Procedures** dialog. To add another, press the **Insert** button. To delete one, press the **Delete** button. Additional buttons allow you to change the order of any procedures listed.

Tip

The purpose of the Procedure button is to add procedures called in embedded source code. The normal way to add template procedures to the Application Tree is to create a menu or toolbar command, add the procedure name via its Actions button, and let the Application Generator automatically add it to the tree.

Embeds

Displays the **Embedded Source** dialog. You can then select either a field specific event or window related action, then add executable source code to customize how the procedure will handle it.

After you choose an embed point in the **Embedded Source** dialog, you choose the code to execute. You can specify a call procedure, which is then added to the tree. You can write your own code with the text editor. Or, you can choose and customize a code template, which is a combination of pre-written code and prompts to "fill-in-the-blanks."

Embedded source gives you complete control over *all* the processing in your procedures. It's one of the most powerful tools Clarion provides you.

Formulas

Accesses the **Formula Editor**, which allows you to create computed and/or conditional fields, which you can then reference in the controls you place in your windows and reports.

Extensions

Accesses extension templates, if any are installed on your system. Extension Templates allow additional functionality through "add-ins" to the Application Generator.

Controls

If any control templates were pre-defined in the current procedure template, or were in a window or report by you, this button accesses the **Action** dialog for the control templates.

Control templates provide "off the rack" controls, such as list boxes, *and* the code to maintain them. This allows you to start with a "bare" procedure template, such as the generic window, and add controls to create your own browse or form windows.

Procedure Dialog

Procedures may call other procedures. Procedure calls specified with template prompts are automatically added to the Application Tree; however, the Application Generator cannot "see" procedures called from embedded source code.

Identifying these embedded procedure calls is important to project management and to source code generation. When you use the **Procedures** button to identify procedures called within embedded source code, the Application Generator can properly display the procedures within the Application Tree hierarchy, *and* the Application Generator can generate the appropriate local MAP structure for the source module.

The **Called Procedure** dialog provides two ways to add procedures:

Selection
This tab control provides a complete list of all procedures defined in the application. Click on any procedure to identify it as a called procedure. You can also use the Select All and Deselect All to mark and unmark all procedures in the list.

List
This tab control provides an alternative method for adding called procedures. This technique is useful when there are too many procedures in your application and the Selection method becomes difficult. In the List section, you manually enter, change, delete and arrange the called procedures using the buttons provided.

Procedure Properties--External

The External Procedure Template declares a procedure is contained in an external library (*.LIB only) or object file. The Application Generator writes no source code. The project system links in the external file as a module.

After selecting the External template type from the **Select Procedure Type** dialog, choose OBJ or LIB from the **Select Module Type** dialog.

Type the file name of the external library or object file in the **Module Name** field. Optionally type parameter declarations in the **Prototype** field.

Description
A short text description for the procedure, which appears next to the procedure name in the **Application Tree** dialog.

Press the ellipsis (...) button to edit a longer (up to 1000 characters) description.

Category
Use this field to group your procedures in the Category Tab of the Application Tree. The Procedure templates add a default value corresponding to their types to group procedures made with the same template together. You can create your own categories as you like.

Prototype
Lets you optionally type a custom procedure prototype which the Application Generator places in the MAP section. See.Prototyping and Parameter Passing with the Application Generator

Module Name
The source code file to hold the code for the procedure. Select from the drop down list. By default, the Application Generator names modules by taking the first five characters of the .APP file name, then adding a three digit number for each module.

The MODULE name for an External procedure should *not* be modified.

Declare Globally
Check this box to generate the procedure's prototype into the PROGRAM's MAP, rather than the MODULE's MAP. This makes the procedure callable from any other procedure, but it also forces a recompile of all program modules whenever you change the prototype.

Export Procedure Declares the procedure in the export file, enabling it to be called by another application. Note: This checkbox is only available when the target file specified in Application Properties as a Dynamic Link Library (.DLL).

Tables This button is not valid for this procedure type.

Window The Window button is disabled for this procedure type.

Report The Report button is disabled for this procedure type.

Data This button is not valid for this procedure type.

Procedures Opens the **Called Procedures** dialog to add (or remove) a procedure to the Application Tree. To add a procedure, press the **Insert** button, then type the procedure name in the next dialog.

Tip

The purpose of the Procedure button is to add procedures called in embedded source code. In all other cases the Application Generator automatically adds procedures to the tree.

Embeds This button is not valid for this procedure type.

Formulas This button is not valid for this procedure type.

Extensions This button is not valid for this procedure type.

Procedure Properties
Window, Browse, Form, Viewer, Menu, ViewOnly

This procedure template is the basis for Window, Browse, Form, Viewer, or Menu procedures. Each of these can add other control or extension templates which add buttons or prompts to this window.

Description
A short text description for the procedure, which appears next to the procedure name in the **Application Tree** dialog.

Press the ellipsis (...) button to edit a longer (up to 1000 characters) description.

Category
Use this field to group your procedures in the Category Tab of the Application Tree. The Procedure templates add a default value corresponding to their types to group procedures made with the same template together. You can create your own categories as needed.

Prototype
Lets you optionally type a custom procedure prototype which the Application Generator places in the MAP section.

Module Name
The source code file to hold the code for the procedure. Select from the drop down list. By default, the Application Generator names modules by taking the first five characters of the .APP file name, then adding a three digit number for each module.

Declare Globally
Check this box to generate the procedure's prototype into the PROGRAM's MAP, rather than the MODULE's MAP. This makes the procedure callable from any other procedure, but it also forces a recompile of all program modules whenever you change the prototype.

Export Procedure
Declares the procedure in the export file, enabling it to be called by another aplication. Note:This checkbox is only available when the target file specified in Application Properties as a Dynamic Link Library (.DLL).

Parameters
Lets you specify parameter names (an optional list of variables separated by commas, with the entire list surrounded by parentheses) for your procedure, which you can pass to it from a calling procedure. You must specify the functionality for the parameters in embedded source code.

Return Value
For functions, lets you specify the variable receiving the return value.

Window Behavior
Press this button to control the behavior of the Window. See Window Behavior

Tables Accesses the Table Schematic Definition dialog. You can define the procedure's access to variables or other files through the dialog.

Window Calls the Window Formatter, to visually design the window.

The ellipsis (...) button next to the **Window** button lets you edit the WINDOW or APPLICATION structure at the source code level. Clarion lets you easily switch back and forth between editing the window graphically, and editing the source code that describes it.

Tip

Take care when hand-editing code for any WINDOW which contains a control template. The Application Generator stores Template Language attributes which cannot be edited by hand (e.g., #ORIG and #LINK).

Report The Report button is disabled for this procedure type.

Data Adds or edits local variables. Press this button and fill in the **Local Data** dialog. Any variables defined are local to the procedure. Define global variables by pressing the **Global** button in the **Application Tree** dialog.

The ellipsis (...) button next to the **Data** button lets you view or edit the memory variable declarations in TXA format.

Tip

You may declare new data items here with normal Clarion language syntax. You do not need to supply the TXA code.

Procedures Opens the **Called Procedures** dialog to add (or remove) a procedure to the Application Tree. To add a procedure, press the **Insert** button, then type the procedure name in the next dialog.

Tip

The purpose of the Procedure button is to add procedures called in embedded source code. In all other cases the Application Generator automatically adds procedures to the tree.

Embeds

Displays the **Embedded Source** dialog. You can then select either a field specific event or window related action, then add executable source code to customize how the procedure will handle it.

After you choose an embed point in the **Embedded Source** dialog, you choose the code to execute. You can specify a call procedure, which is then added to the tree. You can write your own code with the text editor. Or, you can choose and customize a code template, which is a combination of pre-written code and prompts to "fill-in-the-blanks."

Embedded source gives you complete control over *all* the processing in your procedures. It's one of the most powerful tools Clarion provides you.

Formulas

Accesses the **Formula Editor**, which lets you create computed and/or conditional fields, which you can then reference in the controls you place in your windows and reports.

Extensions

Accesses extension and Control templates. Extensions, if applicable to the procedure, can be added or modified from this dialog. If any Control templates were placed in the window, this accesses the prompts for those Control templates. Optionally, you can specify whether or not the prompts for an extension or Control template should display on the procedure properties window.

The **Browse** Template adds several control templates including:

Browse Box control template
Browse Update buttons control template
Browse Select button control template
Close Button control template

The **Form** Template adds:

Save Button control template.
Record Validation extension template.

The **Viewer** Template adds:

ASCII View control template
ASCII Search Button
ASCIIPrint Button
CloseButton

The **ViewOnlyForm** Template adds:

CloseButton
ViewFormActions

The **Menu** Template is nearly equal to the default Window template, but adds a default SDI window with a Menu structure defined.

Procedure Properties—Process

The Process procedure template sequentially processes a data file. You can specify a filter or range of on which records to perform the operation.. A predefined window contains a progress indicator to show the end user what percentage of the operation is complete.

Description
A short text description for the procedure, which appears next to the procedure name in the **Application Tree** dialog.

Press the ellipsis (...) button to edit a longer (up to 1000 characters) description.

Category
Use this field to group your procedures in the Category Tab of the Application Tree. The Procedure templates add a default value corresponding to their types to group procedures made with the same template together. You can create your own categories as you like.

Prototype
Lets you optionally type a custom procedure prototype which the Application Generator places in the MAP section. See.Prototyping and Parameter Passing with the Application Generator

Module Name
The source code file to hold the code for the procedure. Select from the drop down list. By default, the Application Generator names modules by taking the first five characters of the .APP file name, then adding a three digit number for each module.

Declare Globally
Check this box to generate the procedure's prototype into the PROGRAM's MAP, rather than the MODULE's MAP. This makes the procedure callable from any other procedure, but it also forces a recompile of all program modules whenever you change the prototype.

Export Procedure
Declares the procedure in the export file, enabling it to be called by another aplication. Note:This checkbox is only available when the target file specified in Application Properties as a Dynamic Link Library (.DLL).

Parameters
Lets you specify parameter names (an optional list of variables separated by commas, with the entire list surrounded by parentheses) for your procedure, which you can pass to it from a calling procedure. You must specify the functionality for the parameters in embedded source code. See also: PROCEDURE Calls in the Language Reference Manual.

Return Value
For functions, lets you specify the variable receiving the return value.

Process Behavior Press this button to control the behavior of the Window. See Window Behavior

Process Properties Button Press this button to define the process.

General

Window Message The title displayed in the Processing records dialog.

Action for Process The action to perform for each record processed.

No record action specifies no action to performed by the process template. Use embedded source to handle the action.

PUT record specifies that a record will be added.

DELETE record specifies that each record processed will be deleted.

Use RI constraints on action
Check this box to enforce the RI constraints defined in your data dictionary. Clear this box to generate a simple PUT or DELETE depending on the **Action for Process** chosen.

Query Each Deletion Check this box to specify that the user is prompted to confirm each record deletion.

Quick-Scan Records Specifies buffered access behavior for ODBC, ASCII, DOS, or BASIC files. These file drivers read a buffer at a time (not a record), allowing for fast access. In a multi-user environment these buffers are not 100% trustworthy for subsequent access, because another user may change the file between accesses. As a safeguard, the driver rereads the buffers before each record access. To *disable* the reread, enable QUICKSCAN.

Record Filter Type an expression to limit the contents of the browse list to only those records which match the filter expression. This filters all displayable records. When a Record filter is used in conjunction with a Range Limit, only those records within the specified range are filtered.

Tip

You must BIND columns used in a filter expression. See *Hot Fields* below.

Additional Sort Fields Type a comma delimited list of fields on which to sort. These sort fields are in addition to the key for the report set in the **Table Schematic Definition** dialog. If no key is specified, this is the sort order used allowing you to sort records without a key.

Record Count Override When processing in record order (no key), this number is used to calculate what percentage of the operation is complete to provide feedback to the end user. If you don't specify a number, the process "counts" the records before processing begins. This can be relatively fast or slow depending on the file system and the file size. You must supply an appropriate record count when you use a Record Filter (or a Range Limit that results in a filter).

Approx. Record Count This number is displayed in the progress dialog which appears during the process.

Set progress bar limits manually?

Clear this box to make your procedure read the result set and set the progress bar limits automatically. Setting limits automatically may produce poor performance for some SQL data sets, or erratic or inaccurate progress indicator for unevenly distributed result sets. Check this box to manually provide progress bar limits for the procedure. Setting manual limits can provide faster performance for SQL drivers and more accurate progress indicators for unevenly distributed result sets. This setting is only effective if you specify a Key for the File in the **Table Schematic Definition** dialog.

Low Progress Bar Limit

Supply the lowest "free" key element value for the result set. You may type the value or the label of a variable containing the value. Enclose literal string values in single quotes ('value').

High Progress Bar Limit

Supply the highest "free" key element value for the result set. You may type the value or the label of a variable containing the value. Enclose literal string values in single quotes ('value').

Range Limits

Range Limit Field

Type in the field name or press the ellipsis (...) button to select the field from the Component list. The Range Limit Field must be a component of the Access Key specified in the File Schematic dialog. The range limit is key-dependent; the generated source code uses the SET statement to find the first valid record.

Range Limit Type

When a field is selected for **Range Limit Field**, this specifies the method of determining the records for inclusion in the list box.

Current Value -- Signifies the value contained in the key field at the beginning of the ACCEPT loop. This is the value used for the range for the duration of the procedure.

Single Value -- Specifies a variable containing the limiting value. Only records matching the variable are included. Enter a variable in the **Range Limit Value** box which appears, or press the ellipses (...) button to select the variable from the File Schematic.

Range of Values -- Lets you specify upper and lower limits. Enter a variable in the **Low Limit** and **High Limit Value** boxes which appears, or press the ellipses (...) button to select the variables from the File Schematic.

File Relationship -- Lets you choose a range limiting file from a 1:MANY relationship. The Range Limiting field must be the "One" side of a One-to-Many Relationship with the Primary File used in the Process. The relation's linking key must be the same as the Access Key for the procedure. Enter a file in the Related File box, or press the ellipses (...) button to select it from the File Schematic.

Hot Fields

You *must* BIND any variables or EQUATEs used in a filter expression. Press Insert to add the variable to the **Hot Fields** list, and check the **Bind Field** box.

Classes Tab

Use the Classes tab to override the global settings for the Class. See Classes Tab.

Procedure Properties--Buttons

Tables Accesses the Table Schematic Definition dialog. You can define the procedure's access to variables or other files through the dialog.

Window The Window button is disabled for this procedure type.

Report The Report button is disabled for this procedure type.

Data Adds or edits local variables. Press this button and fill in the **Local Data** dialog. Any variables defined are local to the procedure. Define global variables by pressing the **Global** button in the **Application Tree** dialog.

The ellipsis (...) button next to the **Data** button lets you view or edit the memory variable declarations in TXA format.

Tip

You may declare new data items here with normal Clarion language syntax. You do not need to supply the TXA code.

Procedures Opens the **Called Procedures** dialog to add (or remove) a procedure to the Application Tree. To add a procedure, press the **Insert** button, then type the procedure name in the next dialog.

Tip

The purpose of the Procedure button is to add procedures called in embedded source code. In all other cases the Application Generator automatically adds procedures to the tree.

Embeds

Displays the **Embedded Source** dialog. You can then select either a field specific event or window related action, then add executable source code to customize how the procedure will handle it.

After you choose an embed point in the **Embedded Source** dialog, you choose the code to execute. You can specify a call procedure, which is then added to the tree. You can write your own code with the text editor. Or, you can choose and customize a code template, which is a combination of pre-written code and prompts to "fill-in-the-blanks."

Embedded source gives you complete control over *all* the processing in your procedures. It's one of the most powerful tools Clarion provides you.

Formulas

Accesses the **Formula Editor**, which lets you create computed and/or conditional fields, which you can then reference in the controls you place in your windows and reports.

Extensions

Accesses extension and Control templates. Extensions, if applicable to the procedure, can be added or modified from this dialog. If any Control templates were placed in the window, this accesses the prompts for those Control templates. Optionally, you can specify whether or not the prompts for an extension or Control template should display on the procedure properties window.

Procedure Properties--Frame

This template provides an MDI (Multiple Document Interface) parent frame, containing a predefined shell menu. The menu provides useful items such as an Exit command, plus the standard editing and window management commands.

When creating an MDI application, the Frame should be the main procedure. Use the Initiate Thread code template to start new execution threads for each MDI child window which you want to appear inside the frame.

Description A short text description for the procedure, which appears next to the procedure name in the **Application Tree** dialog.

Press the ellipsis (...) button to edit a longer (up to 1000 characters) description.

Category Use this field to group your procedures in the Category Tab of the Application Tree. The Procedure templates add a default value corresponding to their types to group procedures made with the same template together. You can create your own categories as you like.

Prototype Lets you optionally type a custom procedure prototype which the Application Generator places in the MAP section. See.Prototyping and Parameter Passing with the Application Generator

Module Name The source code file to hold the code for the procedure. Select from the drop down list. By default, the Application Generator names modules by taking the first five characters of the .APP file name, then adding a three-digit number for each module.

Declare Globally Check this box to generate the procedure's prototype into the PROGRAM's MAP, rather than the MODULE's MAP. This makes the procedure callable from any other procedure, but it also forces a recompile of all program modules whenever you change the prototype.

Export Procedure Declares the procedure in the export file, enabling it to be called by another application. Note: This checkbox is only available when the target file specified in Application Properties as a Dynamic Link Library (.DLL).

Parameters Lets you specify parameter names (an optional list of variables separated by commas, with the entire list surrounded by parentheses) for your procedure, which you can pass to it from a calling procedure. You must specify the functionality for the parameters in embedded source code. See also: PROCEDURE Calls in the Language Reference Manual.

Return Value For functions, lets you specify the variable receiving the return value.

Splash Procedure Names a procedure to call after the application frame opens, but before any user events are generated. Select from the drop-down list, or type a new procedure name.

By convention, a splash procedure provides a visual or audio (or both) fanfare for your program..

Window Behavior Press this button to control the behavior of the Window. See Window Behavior

Tables Accesses the *Table Schematic Definition* dialog. You can define the procedure's access to variables or other files through the dialog.

Window Calls the Window Formatter, to visually design the window.

The ellipsis (...) button next to the **Window** button lets you edit the WINDOW or APPLICATION structure at the source code level. Clarion lets you easily switch back and forth between editing the window graphically, and editing the source code that describes it.

Tip

Take care when hand-editing code for any WINDOW which contains a control template. The Application Generator stores Template Language attributes which cannot be edited by hand (e.g., #ORIG and #LINK).

Report The Report button is disabled for this procedure type.

Data Adds or edits local variables. Press this button and fill in the **Local Data** dialog. Any variables defined are local to the procedure. Define global variables by pressing the **Global** button in the **Application Tree** dialog.

The ellipsis (...) button next to the **Data** button lets you view or edit the memory variable declarations in TXA format.

Tip

You may declare new data items here with normal Clarion language syntax. You do not need to supply the TXA code.

Procedures Opens the **Called Procedures** dialog to add (or remove) a procedure to the Application Tree. To add a procedure, press the **Insert** button, then type the procedure name in the next dialog.

Tip

The purpose of the Procedure button is to add procedures called in embedded source code. In all other cases the Application Generator automatically adds procedures to the tree.

Embeds

Displays the **Embedded Source** dialog. You can then select either a field specific event or window related action, then add executable source code to customize how the procedure will handle it.

After you choose an embed point in the **Embedded Source** dialog, you choose the code to execute. You can specify a call procedure, which is then added to the tree. You can write your own code with the text editor. Or, you can choose and customize a code template, which is a combination of pre-written code and prompts to "fill-in-the-blanks."

Embedded source gives you complete control over *all* the processing in your procedures. It's one of the most powerful tools Clarion provides you. See also: Adding Embedded Source Code in the Online User's Guide.

Formulas

Accesses the **Formula Editor**, which lets you create computed and/or conditional fields, which you can then reference in the controls you place in your windows and reports.

Extensions

Accesses extension and Control templates. Extensions, if applicable to the procedure, can be added or modified from this dialog. If any Control templates were placed in the window, this accesses the prompts for those Control templates. Optionally, you can specify whether or not the prompts for an extension or Control template should display on the procedure properties window.

Procedure Properties--Splash

The Splash Template generates code to display a window with an image and some text on a 3-D bevelled panel. The window closes automatically after a specified amount of time. In addition, you can optionally allow the user to close the window at any time by CLICKING on it.

Frame procedures are designed to optionally call Splash procedures. Alternatively, you can call Splash procedures with embedded source code.

By convention, a splash procedure provides a visual or audio (or both) fanfare for your program. A splash screen can provide a recognizable logo or icon whose familiarity may raise the user's comfort level and may serve as an advertisement for your program. Additionally it diverts the user's attention from the sometimes boring task of loading and initializing the program.

Description A short text description for the procedure, which appears next to the procedure name in the **Application Tree** dialog.

Press the ellipsis (...) button to edit a longer (up to 1000 characters) description.

Category Use this field to group your procedures in the Category Tab of the Application Tree. The Procedure templates add a default value corresponding to their types to group procedures made with the same template together. You can create your own categories as you like.

Prototype Lets you optionally type a custom procedure prototype which the Application Generator places in the MAP section. See.Prototyping and Parameter Passing with the Application Generator

Module Name The source code file to hold the code for the procedure. Select from the drop down list. By default, the Application Generator names modules by taking the first five characters of the .APP file name, then adding a three digit number for each module.

Declare Globally Check this box to generate the procedure's prototype into the PROGRAM's MAP, rather than the MODULE's MAP. This makes the procedure callable from any other procedure, but it also forces a recompile of all program modules whenever you change the prototype.

Export Procedure Declares the procedure in the export file, enabling it to be called by another application. Note:This checkbox is only available when the target file specified in Application Properties as a Dynamic Link Library (.DLL).

Parameters Lets you specify parameter names (an optional list of variables separated by commas, with the entire list surrounded by parentheses) for

your procedure, which you can pass to it from a calling procedure. You must specify the functionality for the parameters in embedded source code.

Return Value For functions, lets you specify the variable receiving the return value.

Window Operation Mode

This option allows you to override the window settings specified in the **Window Properties** dialog. This allows an additional access point to modify the window's operation mode. See also: WINDOW in the Language Reference Manual.

Use WINDOW Setting specifies no overrides to the window settings

Normal specifies application modal operation mode. The user must respond before moving to any other window in the application.

MDI specifies that the window conform to standard MDI child behavior.

Modal specifies system modal operation. A system modal window takes complete control until the window is closed.

Display Time (in seconds)

Specifies the maximum amount of time the splash window remains displayed.

Close when the user clicks on the splash window

Checking this box lets the user close the window at any time by CLICKING on it.

INI File Settings Checking the **Save and Restore Window Location** specifies that a window's location is stored in the application's .INI file, and will open in that position the next time the procedure is called. This is available only if you enable INI File settings in the **Global Properties** dialog.

Tables Accesses the Table Schematic Definition dialog. You can define the procedure's access to variables or other files through the dialog.

Window Calls the Window Formatter, to visually design the window.

The ellipsis (...) button next to the **Window** button lets you edit the WINDOW or APPLICATION structure at the source code level. Clarion lets you easily switch back and forth between editing the window graphically, and editing the source code that describes it.

Tip

Take care when hand-editing code for any WINDOW which contains a control template. The Application Generator stores Template Language attributes which cannot be edited by hand (e.g., #ORIG and #LINK).

Report The Report button is disabled for this procedure type.

Data Adds or edits local variables. Press this button and fill in the **Local Data** dialog. Any variables defined are local to the procedure. Define global variables by pressing the **Global** button in the **Application Tree** dialog.

The ellipsis (...) button next to the **Data** button lets you view or edit the memory variable declarations in TXA format.

Tip

You may declare new data items here with normal Clarion language syntax. You do not need to supply the TXA code.

Procedures Opens the **Called Procedures** dialog to add (or remove) a procedure to the Application Tree. To add a procedure, press the **Insert** button, then type the procedure name in the next dialog.

Tip

The purpose of the Procedure button is to add procedures called in embedded source code. In all other cases the Application Generator automatically adds procedures to the tree.

Embeds Displays the **Embedded Source** dialog. You can then select either a field specific event or window related action, then add executable source code to customize how the procedure will handle it.

After you choose an embed point in the **Embedded Source** dialog, you choose the code to execute. You can specify a call procedure, which is then added to the tree. You can write your own code with the text editor. Or, you can choose and customize a code template, which is a combination of pre-written code and prompts to "fill-in-the-blanks."

Embedded source gives you complete control over *all* the processing in your procedures. It's one of the most powerful tools Clarion provides you.

Formulas	Accesses the **Formula Editor**, which lets you create computed and/or conditional fields, which you can then reference in the controls you place in your windows and reports.
Extensions	Accesses extension and Control templates. Extensions, if applicable to the procedure, can be added or modified from this dialog. If any Control templates were placed in the window, this accesses the prompts for those Control templates. Optionally, you can specify whether or not the prompts for an extension or Control template should display on the procedure properties window.

Procedure Properties--Source

The Source Procedure template provides an elegant and simple way to add hand code to your application. By default, it provides simple source points at which to embed your code: the data section, code section, local procedures and routines.

The template simply declares the procedure, handles any optional parameters, places the embedded data declarations in the data section, begins the CODE section, then places any embedded executable code in the CODE section:

Description
A short text description for the procedure, which appears next to the procedure name in the **Application Tree** dialog.

Press the ellipsis (...) button to edit a longer (up to 1000 characters) description.

Category
Use this field to group your procedures in the Category Tab of the Application Tree. The Procedure templates add a default value corresponding to their types to group procedures made with the same template together. You can create your own categories as you like.

Prototype
Lets you optionally type a custom procedure prototype which the Application Generator places in the MAP section. See.Prototyping and Parameter Passing with the Application Generator

Module Name
The source code file to hold the code for the procedure. Select from the drop down list. By default, the Application Generator names modules by taking the first five characters of the .APP file name, then adding a three-digit number for each module.

Declare Globally
Check this box to generate the procedure's prototype into the PROGRAM's MAP, rather than the MODULE's MAP. This makes the procedure callable from any other procedure, but it also forces a recompile of all program modules whenever you change the prototype.

Export Procedure
Declares the procedure in the export file, enabling it to be called by another application. Note: This checkbox is only available when the target file specified in Application Properties as a Dynamic Link Library (.DLL).

Parameters Lets you specify parameter names (an optional list of variables separated by commas, with the entire list surrounded by parentheses) for your procedure, which you can pass to it from a calling procedure. You must specify the functionality for the parameters in embedded source code.

Generate Open/Close Files Routines

Check this box to generate an OpenFiles and a CloseFiles ROUTINE for access in your source procedure. Files that you add in the *Other Tables* entry point in the *Table Schematic Definition* dialog will be automatically processed in these ROUTINES. You will still have to explicitly reference these ROUTINES in your hand code (i.e., DO OpenFiles, DO CloseFiles)

Tables Accesses the Table Schematic Definition dialog. You can define the procedure's access to variables or other files through the dialog.

Window The Window button is `disabled for this procedure type.

Report The Report button is disabled for this procedure type.

Data Adds or edits local variables. Press this button and fill in the **Local Data** dialog. Any variables defined are local to the procedure. Define global variables by pressing the **Global** button in the **Application Tree** dialog.

The ellipsis (...) button next to the **Data** button lets you view or edit the memory variable declarations in TXA format.

Tip

You may declare new data items here with normal Clarion language syntax. You do not need to supply the TXA code.

Procedures Opens the **Called Procedures** dialog to add (or remove) a procedure to the Application Tree. To add a procedure, press the **Insert** button, then type the procedure name in the next dialog.

Tip

The purpose of the Procedure button is to add procedures called in embedded source code. In all other cases the Application Generator automatically adds procedures to the tree.

Embeds Displays the **Embedded Source** dialog. You can then select either a field specific event or window related action, then add executable source code to customize how the procedure will handle it.

After you choose an embed point in the **Embedded Source** dialog, you choose the code to execute. You can specify a call procedure, which is then added to the tree. You can write your own code with the text editor. Or, you can choose and customize a code template, which is a combination of pre-written code and prompts to "fill-in-the-blanks."

Embedded source gives you complete control over *all* the processing in your procedures. It's one of the most powerful tools Clarion provides you.

Formulas This button is not valid for this procedure type.

Extensions This button is not valid for this procedure type.

Procedure Properties--Report

This procedure lets you create reports. Press the **Report** button in the **Procedure Properties** dialog to create your report. The procedure template includes a window to show the progress of the report processing. The **Procedure Properties** dialog also includes a checkbox to specify whether you wish to generate a print preview function for your report.

Description
A short text description for the procedure, which appears next to the procedure name in the **Application Tree** dialog.

Press the ellipsis (...) button to edit a longer (up to 1000 characters) description.

Category
Use this field to group your procedures in the Category Tab of the Application Tree. The Procedure templates add a default value corresponding to their types to group procedures made with the same template together. You can create your own categories as you like.

Prototype
Lets you optionally type a custom procedure prototype which the Application Generator places in the MAP section. See Prototyping and Parameter Passing with the Application Generator

Module Name
The source code file to hold the code for the procedure. Select from the drop down list. By default, the Application Generator names modules by taking the first five characters of the .APP file name, then adding a three digit number for each module.

Declare Globally
Check this box to generate the procedure's prototype into the PROGRAM's MAP, rather than the MODULE's MAP. This makes the procedure callable from any other procedure, but it also forces a recompile of all program modules whenever you change the prototype.

Export Procedure
Declares the procedure in the export file, enabling it to be called by another application. Note: This checkbox is only available when the target file specified in Application Properties as a Dynamic Link Library (.DLL).

Parameters
Lets you specify parameter names (an optional list of variables separated by commas, with the entire list surrounded by parentheses) for your procedure, which you can pass to it from a calling procedure. You must specify the functionality for the parameters in embedded source code.

Return Value
For functions, lets you specify the variable receiving the return value.

Report Behavior Press this button to control the behavior of the Window. See Window Behavior

Window Message The text to display displayed in the Progress window.

Tables Accesses the **Table Schematic Definition** dialog. You can define the procedure's access to variables or other files through the dialog.

Window The Window button is disabled for this procedure type.

Report Press this button to call the Report Formatter to visually design the window.

The ellipsis (...) button next to the **Report** button lets you edit the REPORT structure at the source code level. Clarion lets you easily switch back and forth between editing the report graphically, and editing the source code that describes it.

Data Adds or edits local variables. Press this button and fill in the **Local Data** dialog. Any variables defined are local to the procedure. Define global variables by pressing the **Global** button in the **Application Tree** dialog.

The ellipsis (...) button next to the **Data** button lets you view or edit the memory variable declarations in TXA format.

You may declare new data items here with normal Clarion language syntax. You do not need to supply the TXA code.

Procedures Opens the **Called Procedures** dialog to add (or remove) a procedure to the Application Tree. To add a procedure, press the **Insert** button, then type the procedure name in the next dialog.

Tip

The purpose of the Procedure button is to add procedures called in embedded source code. In all other cases the Application Generator automatically adds procedures to the tree.

Embeds

Displays the **Embedded Source** dialog. You can then select either a field specific event or window related action, then add executable source code to customize how the procedure will handle it.

After you choose an embed point in the **Embedded Source** dialog, you choose the code to execute. You can specify a call procedure, which is then added to the tree. You can write your own code with the text editor. Or, you can choose and customize a code template, which is a combination of pre-written code and prompts to "fill-in-the-blanks."

Embedded source gives you complete control over *all* the processing in your procedures. It's one of the most powerful tools Clarion provides you.

Formulas

Accesses the **Formula Editor**, which lets you create computed and/or conditional fields, which you can then reference in the controls you place in your windows and reports.

Extensions

Accesses extension and Control templates. Extensions, if applicable to the procedure, can be added or modified from this dialog. If any Control templates were placed in the window, this accesses the prompts for those Control templates. Optionally, you can specify whether or not the prompts for an extension or Control template should display on the procedure properties window.

Report Properties button

Accesses the **Report Properties** dialog which provides the following tabs and prompts.

Report Template Properties

General

Print Preview

Check this box to enable previewing of a report before printing.

Data Source

Your report can read data to be processed from a target data file (or table), a queue, or from memory.

If you select *File* from the drop list, options for Quick Scan, Record Filter, Additional Sort Fields, Record Count Override, and Progress Bar Limits are available (and discussed below).

If you select *Queue* from the **Data Source** drop list, a **Queue Name** entry is displayed. Press the ellipsis button to the right of the **Queue Name** prompt to select a target queue that the report will read from. The report reads the queue in record sequence, so it will be important to make sure that the queue is built and sorted properly prior to printing.

Finally, if you select *Memory* from the **Data Source** drop list, the report will *not* process (e.g., loop through) the report file designated in the report procedure's Table Schematic. Essentially, this setting is used to print a "snapshot" of the populated report elements and their values when the print process is called.

Quick-Scan Records

Specifies buffered access behavior for ODBC, ASCII, DOS, or BASIC files. These file drivers read a buffer at a time (not a record), allowing for fast access. In a multi-user environment these buffers are not 100% trustworthy for subsequent access, because another user may change the file between accesses. As a safeguard, the driver rereads the buffers before each record access. To *disable* the reread, enable QUICKSCAN.

Record Filter

Type an expression to limit the contents of the browse list to only those records which match the filter expression. This filters all displayable records. When a Record filter is used in conjunction with a Range Limit, only those records within the specified range are filtered.

Additional Sort Fields

Type a comma delimited list of fields on which to sort. These sort fields are in addition to the key for the report set in the **Table Schematic Definition** dialog. If no key is specified, this is the sort order used allowing you to sort records without a key.

Record Count Override

Specify an approximate record count if the report is filtered. The progress indicator uses this value to calculate the percent-completed value.

Set progress bar limits manually?

Clear this box to make your procedure read the result set and set the progress bar limits automatically. Setting limits automatically may produce poor performance for some SQL data sets, or erratic or inaccurate progress indicator for unevenly distributed result sets. Check this box to manually provide progress bar limits for the procedure. Setting manual limits can provide faster performance for SQL drivers and more accurate progress indicators for unevenly distributed result sets. This setting is only effective if you specify a Key for the File in the **Table Schematic Definition** dialog.

Low Progress Bar Limit

Supply the lowest "free" key element value for the result set. You may type the value or the label of a variable containing the value. Enclose literal string values in single quotes ('value').

High Progress Bar Limit

Supply the highest "free" key element value for the result set. You may type the value or the label of a variable containing the value. Enclose literal string values in single quotes ('value').

Report Target

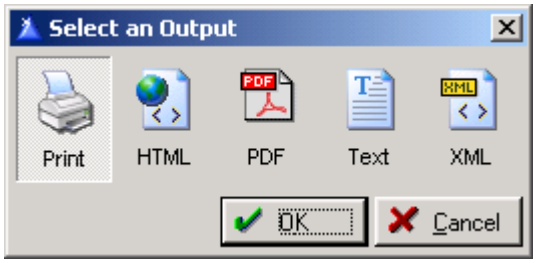
This tab control and its options are only available if you have the Advanced Report Generation templates enabled. The options here allow you to designate a default report target and method to use with these supported templates.

Note:

If this tab control does not appear after including the Advanced Report Generation Global , Extension, make sure to refresh the template generation sequence by selecting **Source** or **Embeds** in the proper procedure.

Report Target

From the drop list, select the initial output the report is to be directed to. If you select *Printer*, the report defaults to a standard printed output. Selecting *Other* enables the **Other Target** prompt, where you can select a different output type by default. Select *Ask at Runtime* to allow a popup window to display before the report begins to process. The window displayed at runtime would look something like this:



Add Print like an option
This option is only available if you have selected *Ask as Runtime* as the report target. Check this box to include standard printer output as a user selectable choice at runtime.

Other target
This drop list is only available if you select Other as your Report Target option. Select from HTML, PDF, TXT or XML as your *default* report target.

Note:

If you have more than one "Report To" template set active, you can always change the default target at runtime in the Print Preview window. Simply select **File > Save As** from the menu, and a list of valid report outputs will be displayed for selection.

Hidden Controls

Each Advanced Report Generation template has the option to hide controls that are not applicable to its output. The idea is to have special controls populated on your report for different types of outputs. For example, you may have a field on a report that stores the proper tag information for a PDF document, and in the same area a hot link tag for HTML. In the Report to HTML template, you can hide the PDF Bookmark field, and in the Report to PDF template, hide the HTML specific control.

The **Hidden Controls** button displays a list box of ALL controls hidden for all available formats that you have enabled. This allows you to quickly search for a control that needs your attention.

Range Limits

Range Limit Field

In conjunction with the **Range Limit Type**, specifies a record or group of records for inclusion in the report. Choose a field by pressing the ellipsis (...) button. The Range Limit Field must be a component of the report's Access Key. The range limit is key-dependent; the generated source code uses the SET statement to find the first valid record.

Range Limit Type

Specifies the type of range limit to apply. Choose one of the following from the drop-down list.

Current Value signifies the current value of the **Range Limit Field**.

Single Value lets you limit the list to a single value. Specify the variable containing that value in the **Range Limit Value** box which appears.

Range of Values lets you specify upper and lower limits. Specify the variable containing the values in the **Low Limit** and **High Limit Value** boxes.

File Relationship lets you choose a range limiting file from a 1:MANY relationship. This limits the report to display only those child records matching the current record in the Parent file. For example, if your report is a list of Orders, you could limit the output to only those orders for the current Customer (in the Customer file).

Preview Options

The Preview Options tab lets you control the initial appearance of the report preview window. This tab is only available if you check the **Print Preview** box on the General tab.

Tip

Although nearly all font types used on reports will print accurately, for the best display of reports in the Print preview window, use a True Type font (Example – Arial 10pt).

Runtime Skip Preview

Press the ellipsis button to select a variable that the template will use to determine whether or not a Print Preview window will be displayed prior to printing the report. A non-zero value will cause the Print Preview to be skipped. A value of zero (or FALSE, the template default) will enable the Print Preview window.

Initial Zoom Setting

Sets the initial magnification for the report to one of four discreet magnification choices. The end user may change the initial setting.

Allow User Variable Zooms?

Check this box to let the end user set custom report magnifications in addition to the preset magnification choices.

Set Initial Window Position

Check this box to enable the four following prompts to set the initial preview window position and size.

X Position	The initial horizontal position of the left edge of the window.
Y Position	The initial vertical position of the top edge of the window.
Width	The initial width of the window.
Height	The initial height of the window.

Maximize Preview Window

Check this box to initially maximize the preview window. This supersedes the **Set Initial Window Position**, whose coordinates are applied only when the window is restored to its normal unmaximized state.

Breaks

The **Breaks** tab provides support for adding *embedded break logic* that is not associated with any bands or break structures on the report. The two uses for this type of logic are used in totaling and conditional headers and footers. The breaks defined here are solely based on field (column) values, and are not associated with any particular band (as in the traditional breaks).

Press the INSERT button to add a new break. You can also change or delete an existing break, or change the break order by using the appropriate move up/move down buttons.

Note:

Each break that is created generates embed points that the developer can use to initialize (*Take Start*), analyze (*Update Total*), and reset (*Take End*) each program break. These embeds are provided by the Break Manager Class, which is the internal logic that supports the template interface. Press the **Embeds** button, and search for *Break Manager* to locate these embed points.

In the Break dialog window, the following prompts are provided:

Name:

Enter a descriptive name for the embedded break that you are creating.

New Break Level 0?

Embedded breaks can be nested. For example, one break can be a totaling break whose result is used in a second totaling break. Check this box if the break will not be dependent on another break.

Fields

Fields are the elements that trigger break logic. Each time the field contents changes, and embedded break is triggered. Additional fields may be added, and *any* field whose contents changes will generate the break.

Totaling

Each time that an embedded break is generated, a total result may be calculated. Press the **Insert** button to create a total field that will be calculated when the break is triggered.

Target Field

Press the ellipsis button to select a **target field** that will hold the totaling result.

Type

From the drop list, select *Count*, *Sum*, or *Average* to determine the type of totaling that need to be performed on the break.

Source Field

Press the ellipsis button to select a **source field** that will be used as the totaling value.

Do Total On

Specify that the total will be calculated on *All Records*, or a *Condition* that is specified below.

Condition

If your **Do Total On** choice is based on *Condition*, enter an expression here that will only include the **Source Field** value if the condition evaluates to “TRUE”.

Reset on Break?

Check this box to reset the **Target Field** after the break has been generated.

Header and Footer

An embedded break can be used to generate a conditional header or footer on a page. An example of this might be a multi-page invoice, where the header of the first page may differ from the header of the second page. In this case, you can create an extra detail section that is used as your “Page Two”, and triggers on a page number embedded break.

Print Header?

Check this box if you would like to generate a header detail when an embedded break is detected. Select a detail band to use for the generated **Header.**

Print Footer?

Check this box if you would like to generate a footer detail when an embedded break is detected. Select a detail band to use for the generated **Footer**.

Hot Fields

You *must* BIND any variables or EQUATEs used in a filter expression. Press **Insert** to add the variable to the **Hot Fields** list, and check the **Bind Field** box.

Detail Filters

Each Detail band is listed in the Detail Filters list. To restrict printing of a band, highlight it in the list, then press the Properties button. Provide an expression in the Filter field. The band will only print when this expression is true.

Optionally, check the **Exclude unfiltered** box to restrict any other detail band which does not have a filter expression of its own.

Classes Tab

Use the Classes tab to override the global settings for the Class. See Classes Tab.

Table (File) Schematic Definition Dialog

You define the files, fields, and variables a procedure--or a control inserted by a control template--can access with the **Table Schematic Definition** dialog. The available data files and keys are the ones you define in the data dictionary.

You can "attach" a file to an item in the **Files** list. These represent the current procedure, module, global data, or other files. When you place a control template in a window, you can "attach" the file to the "To Do" item which appears after you place the control template.

When you select a <To Do> item for a control template, you add a "primary" file from among all the files in your data dictionary. The standard templates *automatically* generate code to open, read, write, and close the primary file. When you select an existing control template file, you add a "secondary" file from a list of related files only. The standard templates *automatically* generate code to open, close, and read the *related* records in the secondary files. So, when you add files to a control template, the templates generate all the code necessary to open, read, write, and close both the primary and any secondary files.

Select **OTHER FILES** when you want to access a file on the same thread, that is already open and positioned to the appropriate record, or when you want to hand code the file access. The templates automatically generate code to open (if not already open) and close **OTHER FILES**, but any other processing is up to you.

The dialog contains the following buttons:

Insert	Allows you to pick a file, previously defined in the data dictionary, from the **Insert File** dialog.
Delete	Deletes the currently selected file from the list (not from the data dictionary, or from disk).
Key	Allows you to select a key previously defined in the data dictionary, from the **Insert Key** dialog. Normally, the file schematic will *automatically* pick the first key listed in the data dictionary for any data file you add by pressing the **Insert** button. This button allows you to pick another key.
New	Allows you to call the Dictionary Editor.

Browse (List Box) Behavior

This chapter describes the fundamental behavior of list boxes within the Clarion IDE, and applies specifically to the template support of scroll bars, locators, and filtering. More specific information regarding the Browse Box template is found later in the Template Guide chapter.

Scroll Bar Behavior

The following section describes the behavior of vertical scroll bars when used with the Clarion IDE template support.

Alpha Distribution

The Alpha distribution divides the scroll bar into 26 segments--one for each letter of the alphabet. Moving the thumb to a segment displays the first record beginning with the segment corresponding letter. For example, moving the thumb to the third segment displays the first record beginning with the letter **c**. If no records begin with **c**, the record with the closest higher value is highlighted.

Last Names distribution

The Last Names distribution divides the scroll bar into 100 segments. Each of these segments is assigned a value based on the distribution of names in an average U.S. telephone book. Each segment is assigned a value and positioning the thumb at that segment displays the first record matching that value. If no records begin with the value, the record with the next closest higher value is highlighted.

The pre-defined Last name distribution is a more accurate method than Alpha when displaying names because names are not evenly distributed alphabetically.

For example, the Browse Box control template uses these values

<null>
ALB
AME
ARN
BAK
BAT
BEN
...

Positioning the thumb on the third segment highlights the first record beginning with the letters AME. If no records match, the next highest next closest higher value is highlighted.

Fixed Thumb

A fixed thumb positions the thumb (or elevator bar) in the center of the scroll bar. Clicking in the scroll bar above the thumb moves up one page at a time. Clicking in the scroll bar below the thumb moves down one page at a time.

Movable Thumb

A movable thumb positions the thumb (or elevator bar) at the top of the scroll bar when the browse box is initialized. Clicking in the scroll bar above the thumb moves up one page at a time. Clicking in the scroll bar below the thumb moves down one page at a time. Draging the the thumb up or down to a position, highlights the closest matching record for that position in the scroll bar. This is dependant on the type of vertical scroll bar behavior you specify.

Custom

This lets you specify the break points for distribution along the scroll bar. This is useful when you have data with a skewed distribution. Insert the values for each point in the list. This divides the length of the scroll bar into segments. Each value you insert in the list creates a segment. For example, if you specify ten values, the scroll bar is divided into ten segments. Positioning the thumb on the third segment highlights the first record beginning with the value of the value of the third item you've inserted in the list. If no records match, the next highest next closest higher value is highlighted.

You can use numeric or string constants, or variables. String constants should be enclosed in single quotes (').

Runtime

The Browse Box is initialized and computes the values for 100 break points based on the first and last record in the Range. The distribution points are determined only when the file is opened; therefore there is no performance penalty. The lowest value in the key is subtracted from the highest value to estimate the range of numbers, 100 evenly distributed points in that range are determined and used to control the vertical scroll bar behavior.

Locator Properties

Step

The user types in a single character to advance the cursor bar in the list box to the record that contains the nearest match in the key field. Use this type of locator only when the first Free Key Element is a STRING, CSTRING, or PSTRING. If no free key element is available, the application generator converts a step locator to None.

Entry Locator

An entry box holds the value for the locator. When the end user places a value in the entry box, pressing TAB or reselecting the list box moves the selection to the nearest matching record. If an entry control is not placed in the window, the application generator converts an Entry locator to a Step locator.

If you use the same field more than once as a locator, you must override the default locator. For example, if you have a multi-keyed browse which has an ascending key and a descending key on the same field. To use separate controls (as on separate TABs) for each condition, check the override box and select the second instance from the drop-down list.

Incremental Locator

When the end user types one or more characters, the list box moves the selection to the nearest matching record. Backspace clears the characters, one-by-one, moving the highlight bar to the nearest matching record of the remaining characters. If a STRING control is placed in the window, the characters display, allowing the user to see the characters as they are entered or cleared. If an ENTRY control is placed in the window, the locator works like an **Entry** Locator when the entry control has focus.

If you use the same field more than once as a locator, you must override the default locator. For example, if you have a multi-keyed browse which has an ascending key and a descending key on the same field. To use separate controls (as on separate TABs) for each condition, check the override box and select the second instance from the drop-down list.

Range Limit Prompts

Using Range Limits and Filters

There are many times that you will want to view, process, or report a sub-set of records from a file. There are two ways to do this:

Range Limit

Filters

Each method has its tradeoffs. Range Limits are much faster to process, but they require that a procedure or control use a limited key as the primary key. Filters are more flexible, since they don't require any special key manipulation, but they are much slower. In any procedure that does sequential processing, you can specify a Range Limit Field, and the type of Range Limit you want to use. The types provided are:

Current Value - Value Limited Keyed Access

The key element specified in the Range Limit Field prompt is the final Fixed Key Element. With this kind of Range Limit, the value of all Fixed Key Elements are saved when the procedure is initialized. These values are used for the duration of the procedure.

Single Value-Value Limited Keyed Access

The key element specified in the Range Limit Field prompt is the final Fixed Key Element. With this kind of Range Limit, the values of all Fixed Key Elements EXCEPT the final Fixed Key Element are saved when the procedure is initialized. The final Fixed Key Element is assigned the value specified in the Range Limit value prompt. This value can be either a variable or fixed value. This value is reevaluated each time a page or entry is loaded or processed.

Range of Values -Ranged Key Access

The key element specified in the Range Limit Field prompt is the first Free Key Element. With this kind of Range Limit, the values of all Fixed Key Elements *except* the final Fixed Key Element are saved when the procedure is initialized. The Low Limit and High Limit Values are used to set the keys for sequential access and to evaluate each record read to insure it is within the valid range. These values can be either fixed values or variables. If variables are used, these variables are reevaluated each time a page or entry is loaded or processed.

There is also support in this dialog for the Higher Key Component, which allows you to specify a secondary key component for **Single Values** and **Range of Values,** and then prime the higher component values accordingly.

Browse Box Only - File Relationship - Value Limited Key Access

All Fixed Key Elements are assigned values as defined in a relationship in the data dictionary. With this kind of Range Limit, it is possible to have multiple Browse Box control templates populated on a window, and as long as the relationships are defined and used, when a parent Browse Box goes out of range, all children (and grandchildren, etc.) Browse Boxes and Controls will automatically be reconstructed.

You *must* BIND any variables or EQUATEs used in a filter expression. Add the variable to the **Hot Fields** list, and check the **Bind Field** box.

Current Value

Signifies the value contained in the key field at the beginning of the ACCEPT loop. This is the value used for the range for the duration of the procedure.

Single Value

Specifies a variable containing the limiting value. Only records matching the variable are included. The variable is reevaluated whenever the window is refreshed. Enter a variable in the Range Limit Value box that appears, or press the ellipses (...) button to select the variable from the File Schematic.

Range of Values

Lets you specify upper and lower limits. Enter a variable in the Low Limit and High Limit Value boxes that appears, or press the ellipses (...) button to select the variables from the File Schematic.

Higher Key Component

The Higher Key Component dialog is used to specify range criteria for key elements that are higher in order than the range key field you have selected as the primary range limit component.

For example, you may have a key defined with a date field and customer number. The key's primary sort (higher component) is the date field, followed by the customer number as the secondary sort. If you choose the customer number as the primary range limit component for your browse, you must also prime the higher key component (date in this example) with a designated value. Failure to do this would cause your range limit to fail on all records, since no date field is zero or NULL.

Based on the key name that you have selected, the higher key component will already be present in this dialog's list box. To modify the default setting displayed, highlight the target component, and press the **Properties** button.

Key Field
This field simply displays the key component you have selected.

Value Type
Choose *Current Value*, *Variable Value*, or *Fixed Value* from the drop list.

Current Value

Primes the key component with the current value stored in the record buffer prior to opening the list box.

Variable Value

Press the ellipsis button next to the **Value** field to assign a variable's value to the selected key component.

Fixed Value

Enter a constant value or expression that will be assigned to the selected key component.

File Relationship

Lets you choose a range limiting file from a 1:MANY relationship. The Range Limiting field must be the "One" side of a One-to-Many Relationship with the Browse Box's Primary File. The relation's linking key must be the same as the Access Key for the Browse Box. Enter a file in the Related File box, or press the ellipses (...) button to select it from the File Schematic.

Fixed Key Element

A Fixed Key Element is an element of a KEY that has a fixed value for the displayable records. For example, when setting a range limit of a single value, all key components up to and including the range limit field have a fixed value. Any key components following the Fixed Key Elements are Free Key Elements.

Database Manager

Database Manager Menu Commands

File Browse Edit Column Project Setup Window Help

The Database Manager allows you direct access to data files without requiring you to create an application. Database Manager thus allows you free access to your data files. The only entry constraint is the picture assigned to a column. For example, if a field has a @n3 picture token, only numbers can be entered. If the picture is changed to @s3, then any character can be entered. This lets you create files for testing purposes.

The Database Manager offers neither data Validity Checking nor Referential Integrity Constraints. This is a programmer's tool; there are no controls to prevent you from making changes that could compromise the integrity of the database.

The following lists the menu commands available from within the Database Manager. Many dialogs also have Help buttons which you can press to view a help topic specifically about that dialog (the F1 key calls the same topic when the dialog is open).

Note that some of the commands, most notably on the Project and Setup menus, do not specifically reference Database Manager functions. Because the Project System and the Registries are always active, their menu commands are always available.

File Menu

New	Opens the **New** dialog, which lets you create a new application file, a new dictionary file, a new source file, or other type of file. You cannot create a new .APP file until you close the current one. You may invoke the **Quick Start Wizard** to help create a new .APP.
Open	Calls the **Open File** dialog, allowing you to open another application, dictionary, source or other file (you must first close the current .APP before opening another).
Pick	Calls the **Pick** dialog, listing the most recently used files by category.
Close	Closes the active data file.
Quick Start New App	Calls the Clarion Quick Start dialog window. You will first access the New dialog window, to designate the name, and where you want to create it.
Save	Saves the active data file.
Save As	Saves the active data file under a new name which you specify.

Save All Saves all currently open files.

Print Prints the active document; for example, a text document is open.

Print Setup Calls the **Printer Setup** dialog, allowing you to configure your printer.

Save As Source Creates a source code document containing a FILE declaration for the current database file.

Convert File Opens the **Convert File** dialog, which creates source code, and a project file for creating an application to convert the data file from one format to another.

File Statistics Opens the **File Statistics** dialog, which provides information about the data file.

Browse Database Browses and edits a database file. Select the file from the **Pick** file dialog, or the **Open File** dialog after specifying a database driver.

Convert Application

The **Convert Application** command starts the Clarion Application Conversion Wizard to convert applications developed in one Clarion environment to another Clarion environment (newer Clarion version, different templates, etc.). See *Application Converter* for more information.

Note:

If your templates are unchanged, you can open application files with newer versions of Clarion and the file conversion is automatic. The Application Conversion Wizard carries out more complex conversions.

Exit Quits the program.

Browse Menu

Order	Opens the **Select File Order** dialog, which lets you choose the active key.
Query by Example	Opens the **Query by Example** dialog, which lets you filter the data file, then display only the records that meet the criteria you specify by entering example values or expressions in this dialog.
Send Driver String	Opens the **Send Driver String** dialog, which lets you execute a SEND command to the file driver.

Edit Menu

Change	Activates an edit control which appears at the selected field and record
Insert	Inserts a blank record at the end of the file, and then activates an edit control which appears in the first field.
Delete	Deletes the current record, first requesting confirmation.
Undelete	Undeletes a record.
Hold	Places a HOLD on the current record.
Release	RELEASEs the previously held record.
Search	Opens the **Search** dialog, which lets you search for the first record containing a value you specify. You may limit the search to one field, or all fields,
Find Next	Repeats the most recent search.
Locate	Opens the **Locate** dialog, which lets you search for the first record containing the value you specify in the key field(s).
Edit Memo	Opens the **Edit Memo** dialog, which lets you edit a memo in ASCII Text.
Hex Edit Memo	Opens the **Hex Edit Memo** dialog, which lets you edit a memo in Hexadecimal format.
OEM Conversion	Specifies string data is converted from OEM ASCII to ANSI when read from disk and ANSI to OEM ASCII before writing to disk.

Column Menu

Hide	Hides the selected column.
Show	Re-displays a previously hidden column.
Picture	Opens the **Picture of Field (fieldname)** dialog, which lets you specify a different display picture for the current column.
Justify	Opens the **Justify Field (fieldname)** dialog, which lets you select a different justification style for the text in the current column.
Reformat	Opens the **Reformat Fields** dialog, which lets you change the field order in the window, and to hide or unhide fields from view..
Header	Opens the **Header Type** dialog, which lets you specify the contents of the header line for the browse window. You can specify, for example, the field label, or the field picture.

Project Menu

Set	Calls the **Open File** dialog, allowing you to change the active .APP or .PRJ.
New	Calls the **New Project File** dialog, allowing you to create a new project.
Load	Opens the **Project Tree** dialog for hand coded projects, or **Application Tree** dialog for generated projects.
Edit	Opens the **Project Tree** dialog, allowing you to add or edit component files in the current project.
Make	Compiles and links the active application or project, which is named on the caption bar.
Make & Run	Compiles, links and runs (if a target executable) the active application or project, which is named on the caption bar.
Run	Executes the active application or project, which is named on the caption bar.
Debug	Loads the Debugger and prepares the active application or project, listed on the caption bar, for debugging.

Make Statistics Calls the **Make Statistics** dialog. Lets you view a statistical profile of the most recent make.

Auto Make Before Run
Toggles the Project System setting which forces a recompile each time you choose the Run command.

File Save Before Run Toggles the Project System setting which saves the source code file each time you choose the Run command.

Minimize on Run Toggles the Project System setting which minimizes CW before displaying the application each time you choose the Run command.

Wait for Termination on Run
Toggles the Project System setting which suspends CW until after you terminate the application upon executing it with the Run command.

Setup Menu

Editor Options Calls the **Editor Options** dialog, which lets you customize the appearance and behavior of the Text Editor.

Dictionary Options Calls the **Dictionary Options** dialog, which lets you specify default settings for the Dictionary Editor.

Application Options Calls the **Application Options** dialog, which lets you specify default settings for the Application Generator.

Template Registry Calls the **Template Registry** dialog, which lets you register and manage templates.

Database Driver Registry
Calls the **Database Driver Registry**, which lets you register database drivers.

Edit Redirection File Loads the Redirection File in a document window, ready for editing.

VCS Options Calls the **Configure VCS Interface** dialog, which lets you configure a selected Version Control System for your dictionaries and applications.

Window Menu

Tile Vertically	Arranges open document windows side by side in a vertical orientation.
Tile Horizontally	Arranges open document windows side by side in a horizontal orientation.
Cascade	Arranges open document windows in overlapped fashion so that all caption bars are visible
Arrange Icons	Arranges iconized windows along the bottom of the Clarion Application frame.
Show Headers	Hides or displays column headers above database fields.
Show Deleted	Hides or displays database records marked for deletion.
Use QBE	This menu toggle lets you enable or disable your QBE filter. When checked, the records displayed match the filter criteria.
(Window List)	Lists all open document windows by their caption bar text according to the order they were opened. Choosing a window from the list brings the window to the top.

Help Menu

Contents	Opens the Windows Help application and displays a list of main topics.
Search for Help On	Opens the **Search** dialog in the Windows Help application, allowing you to search for help topics containing a specific keyword.
How to Use Help	Opens the Windows Help application and displays instructions for using the Help system.
About Clarion	Displays the program name, version, registration, and copyright information.

Edit Memo Dialog

This dialog allows you to edit a memo in ASCII Text. If the file has more than one memo field, you must first select the memo you wish to edit from a list box.

Edit the memo in the text box, then press the **OK** button to return to the view window.

Export File Dialog

This dialog allows you to save a FILE definition for the current data file. You can copy the definition into your source code.

File Label	Type a valid Clarion label for the FILE structure.
Source Filename	Type a DOS file name to save the definition to.

Field List Dialog

This dialog allows you view the basic data for all fields in the data file. You can view the data, but not change it.

The information includes the field label, data type, size, digits, places, and whether the OVER attribute is specified.

Field Picture Dialog

This dialog allows you to specify a new picture token for the currently selected field.

This reformats the way the data displays on screen. This does not alter the data in any way, only the manner in which it is displayed.

File Convert Dialog

This dialog allows you to convert the records in an existing data file to a new file format. When you modify a data dictionary and application, you can use the conversion utility to convert your existing data to the modified format.

The method you use to call the file conversion utility affects its behavior. If you open the converter through the Dictionary Editor (with the appropriate .DCT file open) the converter uses all the information in the dictionary. If start you open a file from any other area, only the information stored in the file header is available. This offers maximum flexibility--allowing you to browse a file without the need for a .DCT.

The information stored in a file header varies according to the file driver.

There are two methods of converting a file--immediate conversion and Generate Source. **Immediate conversion** converts the file once. **Generate Source** creates a source code file, allowing you to make any desired modifications before compiling. Generating and compiling source also creates an executable file that you can ship to end-users to convert their data file.

Before conversion, the utility makes backup copies of the data file and its associated index and memo files. If the conversion process is interrupted, these backup files are renamed to their original names. If you specify a target filename that differs from the original, then the original files are not renamed and are left in place.

The items you specify in this dialog's entry fields controls the conversion.

Source Filename	Specifies the file to convert. This defaults to the file opened by the Database Manager.
Source Dictionary	Specifies the dictionary which contains the file definition for the source data file. If the file conversion utility was invoked from a data dictionary, this defaults to the current dictionary. A Source Dictionary is not required.
Source Structure	Specifies the structure (within the dictionary) which defines the source file. If the file conversion utility was invoked from a data dictionary, this defaults to the current file definition. A Source Structure is not required.
Target Filename	Specifies the name of the new file. This defaults to the current filename.
Target Dictionary	Specifies the dictionary that contains the file definition to which to convert. A Target Dictionary is required.

Target Structure Specifies the structure (within the dictionary) of the target file. The Target Structure is required.

Generated Source The filename for the source code which will create an executable file to change the database. When converting a file, if you want to make any field assignments edit the source code before compiling and executing.

Tip: If you change the name of a field, generate source code, and edit the source code to make the field assignments. Otherwise, your data will be lost. See How to Make a Field Assignment.

For immediate file conversion, without generating source code, see How to Convert a File (without generating source).

File Statistics Dialog

This dialog allows you to examine the file statistics, but not to change them.

Filename	The DOS filename and PATH for the data file.
File Driver	The database driver the file uses.
File Attributes	CREATE, RECLAIM, and ENCRYPT attributes.
Record Length	The size of each record.
Total Number Records	The total number of records in the file (including deleted records).
Number Active Records	The total number of active records.
Deleted Records	The total number of deleted records.
Fields and Field Layout	The number of fields in the file. Pressing the ellipsis (...) button displays the field layout.
Keys and Components	The number of keys in the file. Pressing the ellipsis (...) button displays the key components.
Memos and Layout	The number of memos in the file. Pressing the ellipsis (...) button displays the memo field layout.
Indexes & Components	The number of indexes in the file. Pressing the ellipsis (...) button displays the index components.

Header Type Dialog

This dialog lets you specify the column header for the database fields. Select from the following radio buttons:

Field Label	The column header displays the fields label. This is the default.
Picture	The column header displays the fields picture token.
Type	The column header displays the fields data type.
Group Information	The column header displays the fields group information. If thefield is not part of a group, no header is displayed.
Column Heading	The column header displays the fields column header.
Prompt	The column header displays the fields prompt.

Hex Edit Memo Dialog

This dialog allows you to edit a memo in Hexadecimal format. This is necessary for editing binary format memos. If the file has more than one memo field, you must first select the memo you wish to edit from a list box.

Each character value appears in Hexadecimal format in its own edit box inside the list box. CLICK on the character you wish to change, and type in a new value.

Press the **OK** button to return to the view window.

Justify Dialog

This dialog allows you to specify a new justification style for the currently selected field. This reformats the way the data displays on screen.

Choose the style from the drop down list. Depending upon the field selected, you may choose from Left, Center, Right and Decimal. This does not alter the data in any way, only the manner in which it is displayed.

Locate Dialog

This dialog allows you to search for the first record containing the value you specify in the key field(s). This is only possible when the data file is displayed in a keyed sequence, *not* in Record Number order.

This command only searches fields which are components of the selected key. To search other fields, use the **Search** command.

Pick File Dialog

The Pick dialog is a specialized *Most Recently Used Files* list. As you begin using Clarion for multiple projects, you'll appreciate this dialog because it quickly locates the files you need for any given project.

The Database Manager's **Pick** dialog lists the data files most recently opened for "browsing."

When you choose any of these options, a pick list dialog appears, listing up to twenty of the most recently used files of that type:

The **Pick** dialog provides the following buttons:

Select	Opens the currently selected file.
Remove	Removes the currently selected file from the Pick list.
New	Allows you to create a file.
Open	Allows you to open a file not on the Pick list.
Type	Allows you to change the type of files listed in the **Pick** dialog.

Print Dialog

This dialog allows you to print a record or records.

Current Record	Prints only the currently highlighted record.
Current Page	Prints only the records currently displayed on screen.
All Records	Prints all records in the file.
Use Filter	Prints only those records which match the filter created in the **Query-by-Example** dialog.
Columnar Mode	Prints the records in a "spreadsheet" type of format in which each field in the record is a separate column. Specify the number of records to print side by side in the **Columns** box.
Tabular Mode	Prints the records in a "form" type of format in which each field in the record is on its own separate print line. Specify the maximum width of each field in the **Table Width** box.
Print Header	Specifies whether to print column headers in the report.

Query by Example Dialog

This dialog allows you to filter the data file, then display only the records that meet the criteria you specify by entering example values or expressions in this dialog.

Type the example value in the list box at the top of the dialog, in the column you wish to test. For example, if you want to show all the records where the value of the "Apples" field equals "1," type "1" directly below the "Apples" column header.

To create a query that has the effect of using the AND operator, type a second test value in another field in the same row as the first test value. If, for example, you type "2" directly below the "Cherries" column header, you show all records where the values of the "Apples" field equals "1," and the "Cherries" field equals "2."

To create a query that has the effect of using the OR operator, type a second test value in another row. If for example, you type "3" in the "Apples" column, one row below the "1" in the first query (with no value specified for "Cherries"), you show all records where the values of the "Apples" field equals "1" or "3."

The actual filter expression displays in the group box below the list box as you enter values or logical expressions in the list box. For example, to find all records with an ID number between 10 and 100, with a last name of Smith or Smythe, you create a query:

IDNumber	*FirstName*	*LastName*
>10&<100		='Smith'
>10&<100		='Smythe'

Use the ampersand character (&) to represent the AND operator and the vertical bar (|) to represent the OR operator when used in the same field. The example above can also be represented in this fashion:

IDNumber	*FirstName*	*LastName*
>10&<100		='Smith' | ='Smythe'

Both examples produce a filter expression of (IDNumber > 10 OR IDNumber < 100) AND (LastName = 'Smith' OR LastName = 'Smythe'). The expression displays in the **Filter Expression** group box.

Tip

Although the expression created in a query is not optimized, the runtime evaluator performs its own optimization.

Press the **OK** button to execute the query and display the filtered records in the view window.

Reformat Fields Dialog

This dialog allows you to change the field order in the window, and to hide or unhide fields from view.

The **Shown** list, on the left, lists the fields in the current view. the **Hidden** list, on the right, shows fields *not* in the view. After selecting the fields to hide, show, or move, then pressing the **OK** button, the view window displays the fields you want, in the order you want.

Up	Moves the selected field one position *up* the **Shown** list. This rearranges the view window, moving the field one column left.
Down	Moves the selected field one position *down* the **Shown** list. This rearranges the view window, moving the field one column left.
Hide	Hides the selected field, moving it from the **Shown** list to the **Hidden** list.
Hide All	Hides all fields, moving them from the **Shown** list to the **Hidden** list.
Show	Moves the selected field from the **Hidden** list to the **Shown** list.
Show All	Moves all fields in the **Hidden** list to the **Shown** list.

Search Dialog

This dialog allows you to search for the first record containing a value you specify. You may limit the search to one field, or all fields.

Search	Type the search value.
Exact match	Searches for values that match the specified search string exactly.
Starts With	Searches for values that begin with the specified search string.
Contains	Searches for values that contain the specified search string.
Ends With	Searches for values that end with the specified search string.
Case Sensitive	Specifies case sensitive search testing.
All Fields	Specifies searching all fields in the data file. If not specified, the search is on the currently selected field.

Select Driver Dialog

When first loading a file, the Database Manager prompts you to name the driver used to read the file. Select a previously installed Clarion database driver from the list.

Select File Order Dialog

Once a file is open, you can change the sort order by specifying a different key. This dialog displays a list of available keys, and allows you to change the active key.

Select the key which matches the desired sort order (or Record Order) from the key list, then press the **OK** button.

The file is displayed in the selected sort order, and ready for any Database Manager operation.

Select Memo Dialog

If your data file has more than one MEMO, this dialog appears to allow you to select the MEMO to edit.

Highlight the desired MEMO, then press the **Select** button.

Send Driver String Dialog

This allows you to execute a SEND command to the file driver. Type the driver string in the edit box, and press **OK**.

See the *Database Drivers* PDF for complete information on the SEND commands for each driver.

Dictionary Editor

Dictionary Editor Menu Commands

File Edit Version Project Setup Window Help

The Data Dictionary is the central repository information concerning your application's data. The Dictionary file--.DCT--stores file names, file structures, file relations, file aliases and views, field names, lengths, and data types, field validity checks, field entry pictures, keys, indexes, plus much more such as status bar help messages by field and default prompt values by field.

The Application Generator uses the Data Dictionary to generate source code, such as file declarations, which it places in the data section of the generated source code files. It also uses the dictionary to provide, for example, entry pictures when formatting entry dialogs for the end user.

The following lists the menu commands available from within the Dictionary Editor. Many dialog also have Help buttons which you can press to view a help topic specifically about that dialog (the F1 key calls the same topic when the dialog is open).

Note that some of the commands, most notably on the Project and Setup menus, do not specifically reference Dictionary Editor functions. Because the Project System and the Registries are always active, their menu commands are always available.

File Menu

New	Opens the **New** dialog, which lets you create a new dictionary or other type of file.
Open	Calls the **Open File** dialog, allowing you to open a Dictionary file.
Pick	Calls the **Pick** dialog, listing the most recently used files in a list box.
Close	Closes the active file.
Quick Start New App	Calls the Clarion Quick Start dialog window. You will first access the New dialog window, to designate the name, and where you want to create it.
Save	Saves the active file.
Save As	Saves the active file under a new name which you specify.
Save All	Saves all the active files.
Print	Prints the active document.

Print Setup Calls the **Printer Setup** dialog, allowing you to configure your printer.

Change Directory Opens a Windows **Directory** dialog. Any subsequent file dialogs will default to the selected directory.

Search Files Opens the **File Search** dialog. Allows standard searches for files.

Refresh Dictionary Reapplies all attributes from the parent field to all fields that are derived, and not frozen. See Derived From.

Synchronize Dictionary
Opens the Dictionary Synchronizer.

Create Conversion Program for One File
Opens the **Convert File** dialog, which creates source code, and a project file for creating an application to convert the data file from one format to another.

Create Conversion Program for Multiple Files
Opens the Dictionary Synchronizer to generate a multiple file conversion program.

Import Table Lets you add a file definition from an existing data file to the current dictionary. Just choose the file and driver, in the Import File dialog which this command opens.

Tip

To import a file definition for an ODBC source which stores multiple tables in the same file (such as Microsoft Access), be sure the Data Source is correctly specified in the ODBC.INI file. Then select the Data Source and the table in the dialogs that appear. The import will add all fields except memos.

Multi Table Import Calls the Dictionary Synchronizer to import multiple table definitions.

Import Text Imports a dictionary stored in .TXD (text) format.

Export Text Exports a dictionary stored in .TXD (text) format.

Check In Allows you to check in a selected dictionary to your target Version Control System.

Check Out Allows you to check out a selected dictionary modules from your target Version Control System.

Browse *table* Loads the Database Manager to browse the selected file.

Convert *table* — Loads the Database Manager, and converts the file if needed.

Browse Database — Loads the Database Manager

Convert Application — The **Convert Application** command starts the Clarion Application Conversion Wizard to convert applications developed in one Clarion environment to another Clarion environment (newer Clarion version, different templates, etc.). See *Application Converter* for more information.

Note:

If your templates are unchanged, you can open application files with newer versions of Clarion and the file conversion is automatic. The Application Conversion Wizard carries out more complex conversions.

Exit — Quits the program.

Edit Menu

Cut — Deletes the selected file definition, field, or key from the dictionary and places it in the clipboard.

Copy — Places a copy of the selected field, or key from the dictionary into the clipboard.

Paste — Pastes the previously copied file definition, field, or key from the dictionary from the clipboard into the active dictionary.

Add File — Adds a new file to the active dictionary.

Add Alias — Adds a new alias to the active dictionary.

Properties — Calls the **Properties** dialog for the selected file, alias, field, key, etc., from the active dictionary.

Columns/Keys — Calls the **Columns/Keys Definition** dialog for the selected file.

Triggers — Calls the Triggers tab control located on the Columns/Keys Definition window.

Delete... — Depending on the dialog, deletes the selected file, alias, field, key, etc., from the active dictionary. The actual commands vary according to the buttons in the active dialog which "delete" something to a list in the dialog.

Refresh Table	Reapplies all attributes from the parent field to all fields that are derived, and not frozen for the selected table only. See Derived From.
Distribute Table	Reapplies all attributes from the parent field to all fields that are derived, and not frozen for all tables in the active dictionary.
Add Column	Adds a new column to the active (selected) table.
Column Properties	Calls the **Properties** dialog for the selected column.
Delete Column	Deletes the selected column.
Move Column Up	Moves the selected column shown up in the list.
Move Column Down	Moves the selected column shown down in the list.
Add Relation	Adds a new file relation to the active dictionary.
Relation Properties	Calls the **Properties** dialog for the selected file relation.
Delete Relation	Deletes the selected file relation from the active dictionary.
Dictionary Properties	Opens the **Dictionary Properties** dialog for the active dictionary.
View as Toolbox	Opens the Dictionary Toolbox for the application's data dictionary. The Dictionary toolbox provides a hierarchical list representing your database. This list shows database files, keys, key components, fields, and relationships in an expanding hierarchical tree. The Dictionary toolbox can update all the *existing* items in the application's data dictionary except the dictionary properties.

Version Menu

Checkpoint

Adds one revision step to the current dictionary.

The Dictionary Editor automatically places an internal version number in your dictionary file. A new dictionary automatically starts with version 1.0. You can see the version number/revision number on the caption bar of the **Dictionary** dialog. The **Dictionary Properties** dialog also displays the original creation date and time, and the last modified date and time.

You should increase the version number, manually, whenever you make significant changes to a dictionary; for example, when you're working on version #2 of your application. The revision number (r. #) on the caption bar increases by one.

Revert

Rolls back to a previous version. Choose the revision to revert to by selecting it with the spin control in the **Previous Revision** dialog.

Project Menu

Set

Calls the **Open File** dialog, allowing you to change the active .APP or .PRJ.

New

Calls the **New Project File** dialog, allowing you to create a new project.

Load

Opens the **Project Tree** dialog for hand coded projects, or **Application Tree** dialog for generated projects.

Edit

Opens the **Project Tree** dialog, allowing you to add or edit component files in the current project.

Make

Compiles and links the active application or project, which is named on the caption bar.

Make & Run

Compiles, links and runs (if a target executable) the active application or project, which is named on the caption bar.

Run

Executes the active application or project, which is named on the caption bar.

Debug

Loads the Debugger and prepares the active application or project, listed on the caption bar, for debugging.

Make Statistics

Calls the **Make Statistics** dialog. Lets you view a statistical profile of the most recent make.

Auto Make Before Run
Toggles the Project System setting which forces a recompile each time you choose the Run command.

File Save Before Run Toggles the Project System setting which saves the source code file each time you choose the Run command.

Minimize on Run Toggles the Project System setting which minimizes CW before displaying the application each time you choose the Run command.

Wait for Termination on Run
Toggles the Project System setting which suspends CW until after you terminate the application upon executing it with the Run command.

Setup Menu

Editor Options Calls the **Editor Options** dialog, which lets you customize the appearance and behavior of the Text Editor.

Dictionary Options Calls the **Dictionary Options** dialog, which lets you specify default settings for the Dictionary Editor.

Application Options Calls the **Application Options** dialog, which lets you specify default settings for the Application Generator.

Template Registry Calls the **Template Registry** dialog, which lets you register and manage templates.

Database Driver Registry
Calls the **Database Driver Registry**, which lets you register database drivers.

Edit Redirection File Loads the Redirection File in a document window, ready for editing.

VCS Options Calls the **Configure VCS Interface** dialog, which lets you configure a selected Version Control System for your dictionaries and applications.

Window Menu

Tile Vertically	Arranges open document windows side by side in a vertical orientation.
Tile Horizontally	Arranges open document windows side by side in a horizontal orientation.
Cascade	Arranges open document windows in overlapped fashion so that all caption bars are visible
Arrange Icons	Arranges iconized windows along the bottom of the Clarion Application frame.
(Window List)	Lists all open document windows by their caption bar text according to the order they were opened. Choosing a window from the list brings the window to the top.

Help Menu

Contents	Opens the Windows Help application and displays a list of main topics.
Search for Help On	Opens the **Search** dialog in the Windows Help application, allowing you to search for help topics containing a specific keyword.
How to Use Help	Opens the Windows Help application and displays instructions for using the Help system.
About Clarion	Displays the program name, version, registration, and copyright information.

Columns/Keys/Triggers Definition Dialog

The **Columns/Keys/Triggers Definition** dialog contains three tabs--listing **Columns**, **Keys**, and **Triggers** respectively.

- *To add a new column*, select the **Columns** tab, then press the **Insert** button. The **New Column Properties** dialog appears.
- *To add a new key*, select the **Keys** tab, then press the **Insert** button. The **New Key Properties** dialog appears.

Columns

- *To modify an existing column,* select it and press the **Properties** button. The **Edit Column Properties** dialog appears.
- *To delete an existing column*, select it and press the **Delete** button.
- *To move the selected column within the Columns list,* press the ↑ and ↓ buttons. This reorders the column labels within the FILE structure.

Keys

- *To modify an existing key or key component*, select it and press the **Properties** button. The **Edit Key Properties** dialog appears.
- *To delete an existing key*, select it and press the **Delete** button.
- *To move the selected key within the Keys list,* press the ↑ and ↓ buttons.

Keys and indexes specify sort orders for a single table. A key may reside within the table, or as an external file, depending on the database. Keys are automatically updated whenever records are added, changed, or deleted. See *Database Drivers* for more information regarding how each database supports keys or indexes.

Indexes usually exist as external tables. Remember that a separate DOS file handle is necessary for each external key or index table. Index tables do *not* update automatically. The BUILD statement updates an index.

A dynamic index allows you to declare an index table without specifying the column(s) in the Data Dictionary. The application must define the column(s) at runtime, as the second parameter of the BUILD statement. The application may rebuild the same index table at a later time, specifying a different column for the index key.

Data Sources dialog

Select the data source from which the table will be imported, then press the **Next** button to select the table to import. Select the table, then press the **Finish** button to import the table.

If the Data Source has not been defined in ODBC.INI, press the **New** button to define the data Source. This calls the ODBC Administrator, an applet which maintains table definitions in ODBC.INI.

Dictionary Dialog

Allows you to manage the data tables, aliases, and relations for your dictionary.

Tables List The list of data tables and aliases in the dictionary.

Add Table Pops up a sub menu that lets you add a new table, alias, or import a new table from an external source.

Properties Allows you to change the options for the selected table or alias.

Columns/Keys Allows you to define columns and keys for the selected table or alias.

Delete Deletes the selected table or alias.

Browse Calls the Database Manager to browse the selected table or alias.

Convert Converts the physical format of the selected table to match the data dictionary table definition.

Use the locator to the right of the Convert button to help you quickly locate a table. Useful in large dictionaries.

Columns List Shows columns for the currently selected table or alias.

Add Column Allows you to add a new column to the selected table shown in the *Tables List*.

Properties Allows you to change the options for the selected table or alias. See also: Edit Properties dialog.

Delete Deletes the selected column.

Use the locator to the right of the **Delete** button to help you quickly locate a table. Useful in large dictionaries.

Related Tables List

Shows relations for the currently selected table or alias.

Add Relation	Allows you to add a relation to the selected table or alias.
Properties	Allows you edit the selected relation.
Delete	Allows you to delete the selected relation.
Display Key List	Press this icon button to toggle this display to the **Keys List**

Keys List

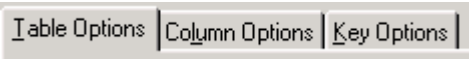
Shows keys for the currently selected table or alias.

Add Key	Allows you to add a key to the selected table or alias.
Properties	Allows you edit the selected key.
Delete	Allows you to delete the selected key.
Display Relation List	Press this icon button to toggle this display to the **Relations List**

Other Buttons

Dictionary Properties	Allows you to add or edit information about the current data dictionary, including creation, modification dates, and a text description.
Synchronize	Calls the Dictionary Synchronizer (available only in Enterprise Edition). See Dictionary Synchronizer Overview for more information.

Dictionary Options Dialog



You can customize the default dictionary settings in this dialog. To access the dialog, choose **Setup ▸ Dictionary Options**.

Table Options

Default Driver To select the default database driver for new tables in a dictionary, choose from the drop down list.

For detailed descriptions of the drivers available, see *Database Drivers*.

Sort Dictionary Tables Alphabetically
Check this box to display tables in alphabetical order. If not checked, tables display in the order in which they were created.

Show Key List The lower right pane of the main Dictionary Editor window switches display between Keys and Relations. Check this box to defauly the display to Keys, or check it off if you would like to view Relations on start up.

Default THREAD Attribute
To specify new table definitions default to adding the THREAD attribute (setting aside a separate RECORD buffer for each procedure), check this box.

Default OEM Attribute
To specify new table definitions default to adding the OEM attribute (set International string support), check this box.

Display Table Description
Check this box to display the table description in the Tables list.

Display Table Driver Check this box to display the table driver in the Tables list.

Display Table Prefix Check this box to display the table prefix in the Tables list.

Quick Load

Prompt to Use Quick Load
Check this box to always be prompted to use the Quick Load dictionary feature when creating new tables. If checked off, you will always enter the *Table Properties* window when selecting **Add Table.**

Use for new table Check this box to automatically use the Quick Load feature when selecting the **Add Table** option.

Column Options

Assign description to message
Check this box to specify that the column descriptions you specify when defining a column should automatically serve as the text for the MSG attribute (setting a status bar message when controls referencing the column have the focus).

Assign description to tooltip
Check this box to specify that the column descriptions you specify when defining a column should automatically serve as the text for the TIP attribute (setting the tool tip message when controls referencing the column have the focus).

Display Column Description
Check this box to display the column description in the Columns list.

Display Column Type Check this box to display the column's data type in the Columns list.

Display Column Picture
Check this box to display the column's display picture in the Columns list.

Display Column Prefix
Check this box to display the column's prefix in the Columns list.

Key Options

Display Key Description
Check this box to display the Key description in the Keys list.

Display Key Type Check this box to display the Key Type in the Keys list.

Display Unique Flag Check this box to display **Unique** if the Key is flagged as unique.

Display Primary Key Status
Check this box to display **Primary** if the Key is flagged as the Primary Key.

Display Other Key Attributes
Check this box to display the other attributes of the Key in the Keys list.

Display Key Prefix Check this box to display the Key prefix in the Keys list.

Dictionary Properties Dialog

Displays information about the current data dictionary, including creation, modification dates, and a text description. Press the **Password** button to password protect your dictionary from modification by other users.

Properties

Created	The original file creation date.
Last Modified	The most recent modification date. See also: **Version ▸ Checkpoint**; **Version ▸ Revert** menu options.

Comments

Allows you to enter a text description describing the dictionary. The description is solely for your convenience, and has no effect on the application. It is useful for situations in which other programmers may pick up your code later, or for when you expect to return to the project after a long period of time since you last looked at it.

Password

Calls the **Password Validation** dialog which enables password protection for your dictionary

Dictionary Triggers

The Triggers interface of the Dictionary Editor allows you to enter Clarion source that can be applied before or after any data operation is performed on a selected file. Press the **Insert** button to add a dictionary trigger.

A trigger can include any Clarion language statement to execute as a unit and can also invoke stored procedures defined in your application. However, procedures and triggers differ in the way that they are invoked. A procedure is explicitly executed by a user, application, or trigger. Triggers (one or more) are implicitly fired (executed) by the supported database when a triggering INSERT, UPDATE, or DELETE statement is issued, no matter which user is connected or which application is being used.

Uses of Triggers

Triggers can supplement the standard capabilities of a database to provide a highly customized database management system. For example, a trigger can restrict operations against a table to those issued during regular business hours. A trigger could also restrict operations to occur only at certain times during weekdays. Other uses for triggers are to

- automatically generate derived column values
- prevent invalid transactions
- enforce referential integrity across nodes in a distributed database
- enforce complex business rules
- provide transparent event logging
- maintain synchronous table replicates
- gather statistics on table access
- store complex computed and conditional fields that are table related in one location

Triggers are useful for customizing a database. However, you should use triggers only when necessary. The excessive use of triggers can result in complex interdependencies, which may be difficult to maintain in a large application. For example, when a trigger is fired, a statement within its trigger action potentially can fire other triggers.

Dictionary Version Control

Version ▸ CheckPoint

Increases the internal version number in the data dictionary.

The Dictionary Editor automatically places an internal version number in your dictionary file. A new dictionary begins with version 1.0. You can see the version number/revision number on the caption bar of the **Dictionary** dialog. The **Dictionary Properties** dialog also displays the original creation data and time, and the last modified date and time.

You should increase the version number, manually, whenever you make significant changes to a dictionary; for example, when you're working on version #2 of your application, choose **Version ▸ Checkpoint**. The revision number (r. #) on the caption bar increases by one.

To roll back to a previous version, choose **Version ▸ Revert**. Choose the revision to revert to by selecting it with the spin control in the **Previous Revision** dialog.

Import Table Dialog

The Dictionary Editor allows you to quickly add a data table to the dictionary by creating a data definition based on an existing data file.

With the Dictionary dialog active, select File > Import Table. Specify a data table and additional options in the **Import Table** dialog.

Filename	Type a file name or press the ellipsis (...) button to select a name from the **Open *driver* file** dialog. When importing a definition via an ODBC data source, do not specify a directory name; Clarion will read it from ODBC.INI.
File Driver	Choose a database driver from the drop down list.
Owner	Fill in an optional OWNER attribute. If importing from an ODBC data source, type in the datasource name, (optional) UserID, and (optional) Password.
Options	Fill in any optional driver strings.

After reminding you that the import process cannot import memo fields (you can add them manually), the data table is added to the dictionary, along with its column and key definitions. When importing from an ODBC data source, no key definitions will be imported.

Dictionary Wizatron

(Only available in Enterprise and Web Editions)

The purpose of the Dictionary Wizatron is to prime an active data dictionary with User Options that are recognized and processed by the subsequent application and procedure templates. Using the Dictionary Wizatron with your active dictionary ensures that the application wizatron generates by default a solid application with built-in data integrity.

The most important control to understand when navigating through the *Wizard* view is found near the bottom of the screen. It is a slider control called the **Prompt Detail.**

There are incremental settings on the Prompt Detail found in the Dictionary Wizatron. The current setting (slider to the far left) reflects the **Minimum Detail**. This setting identifies the minimum required information that the Dictionary Wizatrons deems important.

The various input controls located above the Prompt Detail are the actual *prompts* used by the Wizatron to gather the necessary information needed to populate the dictionary with appropriate user options. These prompts can be standard entry controls, drop controls with a fixed list of choices, or lookup (ellipsis) controls used for a variety of selection options.

Auto-Increment and OVER settings

The first window of the Dictionary Wizatron controls how to treat all Auto-Incrementing columns and their keys, and provides population options for columns defined with an OVER attribute (the Place Over option in the dictionary).

For all auto number keys, set the “Do Not Auto-Populate This Key” option for

Specifies to populate incrementing, non-incrementing, or entire key elements that are components of any auto-numbering key.

Column Interest Level

Column Interest allows you to override the default interest values set by the Dictionary Wizatron. All dictionary elements have a relative interest level that are detected by other programs using the **INTEREST** User Options symbol. Interest levels range from 1 to 10. Here are some recommended default conditions:

Data elements assigned low interest levels (1-4) normally should be sparsely populated or not seen at all by your application.

Data elements not modified by the developer are given a default value of five(5), considered “mid-range” and receive no special attention than they normally would. They are considered neutral.

Interest levels from 6-10 normally imply that these data elements special treatment. They are populated and appear more frequently in your application.

It would be a tedious task to assign every data element in your dictionary a specific interest level. The Dictionary Wizatron allows you to enter a partial value here that populates any column name matching it.

Controlling Third Normal Form

In the third normal form of database normalization, duplicate information is eliminated.

For example, if two tables in a Clarion dictionary require an author name column, the author name is stored in a separate table. The author name column is accessed via a unique AuthorID column and associated key, stored in the table containing the author name. More importantly, any change to this author name is automatically reflected in all tables that link (have relationships) with the Author table.

This part of the Dictionary Wizatron supports the third normal form in your applications. By checking the interest levels of the columns in the related tables, the Wizatron calculates which related information is important from the lookup table, and alerts the corresponding template to substitute the appropriate columns in designated procedures. This is accomplished using the DISPLAY option in the User Options of the related link column.

The **Substitute Column Interest** spin control causes the Wizatron to generate the DISPLAY option into those linking columns that have an interest level greater than or equal to the value entered here.

The **Max number of columns** spin control specifies the number of columns that will be populated by the DISPLAY option.

The **Modify Interest Level** check box modifies the lookup columns' interest level (normally of low interest) to be the average of the substituted columns.

Edit-In-Place Options

This window provides the ability to enable edit-in-place updates to a browsed table instead of a separate update (form) procedure. This option is recommended for tables with one-way lookup relationships, such as a State Code table. Tables with complex relationships are better managed with a separate update procedure. The Dictionary Wizatron simply seeds the target table with the **EDITINPLACE** User Option.

Other Key File Calculations

Calculate Interest Levels
This feature calculates the interest level of any "populating" key based on the interest level of the key fields.

Calculate "Text Primary" key.
This forces the key with the highest interest level to be assigned the **TEXTPRIMARY** option. The key with the **TEXTPRIMARY** option becomes the primary sort key of wizard generated browses.

Prompts and Headers

The last option screen allows you to take full advantage of the prompt and header information that is stored in the data dictionary. For example, the lookup fields that you designate via the DISPLAY User Options will often override the default linking columns' prompts and/or column headers with their own values.

Remember that you have total control of the prompts and header information that the wizards generate. This option allows you to fine tune this information.

This screen provides a relational tree view of your data dictionary, with edit-in-place options for prompts and headers of the individual columns.

At the completion of the Dictionary Wizatron, there is a summary window that shows a complete listing of column labels and the action that was performed on them.

If you need a copy of the results, press the **Copy** button. This will send your results to the Windows clipboard. You can later paste the results into the text editor of your choice for saving and/or printing.

New/Edit Column Properties Dialog

The **New Column Properties** dialog allows you to define columns and variables, and to set column or variable related options and attributes.

All the Clarion language attributes that you can place on a column also apply to memory variables. There are only a few additional attributes that can *only* be placed on global or local memory variables. These are disabled when defining a column.

The Dictionary Editor allows you to add the columns one after another, quickly. Each time you complete and close the **New Column Properties** dialog for one column, another blank **New Column Properties** dialog appears, ready for the next column. Press **Cancel** when the blank dialog appears after completing the last column, to return to the **Column/Key Definition** dialog.

General

Column Name

To name the column, type a valid Clarion label in the **Name** prompt. Valid column names may vary slightly according to the file driver.

Derived From

Press the ellipsis button (...) to select another (parent) column in the dictionary from which to copy all column attributes, except the column name. The parent column may be any other column in the data dictionary, including global data, column pool, or file columns.

Press the refresh button to reapply the attributes from the parent column. Use the **Freeze** check box below to prevent a refresh from the parent. See also the **Refresh Column, Refresh Table, Refresh Dictionary,** and **Distribute Column** menu commands.

Description

To add a text description, type it in the **Description** field. The description appears next to the column name in various dialogs. You can optionally assign the description to the MSG attribute by choosing **Setup ▸ Dictionary Options** and checking the appropriate box.

Tip

Clarion's Wizards use this description when creating Browses, Forms, and Reports.

Data Type

To assign a column type, choose one from the **Type** drop down list. Clarion supports the following column types, which specify how the data will be stored on disk by the file driver, and accessed in memory by the application. These correspond to the Clarion variable types, plus memo and picture types. The types available vary according to the selected database driver. See the Language Reference Manual for a complete list of data types available.

Tip

The Decimal type generally provides the best all around performance for mathematical calculations. The compiler optimizes the operation by multiplying values by powers of ten before processing; this greatly speeds up performance on systems without math coprocessors, at no cost in mathematical precision. See also: BCD Operations and Procedures in the Language Reference manual.

Base Type

Specify the label of a user-defined data type. The user-defined data type can be a GROUP, a QUEUE, or an object. See TYPE.

For example:

```
G1 GROUP,TYPE !a user-defined data type
S1 STRING(10)
S2 STRING(10)
END
```

Use the **Base type** column to specify the G1 label so the Application Generator generates something like the following:

```
MyTypeField G1
```

Reference

To create a reference variable, check the **Reference** box. A reference variable stores the memory address of another variable. This box is enabled only when defining memory variables

Binary

To specify that a MEMO field may hold binary data, check the **Binary** box. This is dependent on the file driver. This adds the BINARY attribute.

Characters

To assign a column length specify a number in the **Chars** field.

Places

To assign a set number of decimal places for a real number, specify a number in the **Places** field.

Dimensions

To declare the variable as an array, and to specify the array dimensions, type them in the **Dimensions** fields. You can specify up to four dimension sizes. This adds the DIM attribute.

Record Picture

To specify the picture for a picture column, type it in the **Record Picture** field.

Screen Picture

To specify a screen picture , type it in the **Screen Picture** field.

"Lock" icon

To lock the screen picture, which specifies that it may not be changed if the column type is changed, press the "Lock" icon next to the **Screen Picture** field.

Prompt Text

To specify the default prompt string, type it in the **Prompt Text** field. The Application Generator uses this for controls which display an on screen prompt.

Column Heading

To specify the default column title, type it in the **Column Heading** field. The Application Generator uses this for reports.

Freeze

Check this box to prevent column attribute refresh from the parent column. Use the **Derived From** field above to set the parent column. See also the **Refresh Column, Refresh Table, Refresh Dictionary,** and **Distribute Column** menu commands.

Attributes

Case

To specify the case attribute for controls referencing the column, choose from the **Normal**, **Capitals** or **Uppercase** radio buttons, in the **Case** group box. The Application Generator adds the CAP or UPR attributes.

Typing Mode

To specify the default typing mode attribute for controls referencing the column, choose from the **Insert**, **Overwrite** or As Is radio buttons in the **Typing Mode** group box. The Application Generator adds the INS or OVR attributes.

Flags

Immediate

To specify immediate event notification for controls referencing the column, check the **Immediate** box. The Application Generator adds the IMM attribute.

Password

To specify the data non-display attribute for controls referencing the column, check the **Password** box. The Application Generator adds the PASSWORD attribute. When an end user types in an entry control referencing this column, the characters typed do not appear on screen.

Read only

To specify the display only attribute for controls referencing the column, check the **Read only** box. The Application Generator adds the READONLY attribute.

Justification

To specify justification for controls referencing the column, select from the **Alignment** drop down list. The Application Generator adds the LEFT, RIGHT, CENTER or DECIMAL attribute.

Offset

To specify an indentation amount for controls referencing the column, specify a number in the **Indent** field. The Application Generator uses this setting as the parameter for the LEFT or RIGHT attribute. The measurement unit depends on the default measurement unit for the window in which a control referencing the column resides.

Initial Value

To specify a default value for the column, type it in the **Initial Value** field. Specifying an initial value for a database column generates an assignment statement.

- A literal character value enclosed in single quotes generates:

```
STATE:Code    = 'FL'
```

- For numeric data types, a number without quotes generates:

```
CUS:CreditLine = 1000
```

- The label of a global variable or a local variable that is visible to the procedure generates:

```
USER:Preference     = MyGlobalPreference
```

- A function generates:

```
INV:Date = TODAY()
```

Specifying an initial value for a global, module, or local memory variable generates a data declaration statement.

- A literal character value without quotes generates:

```
MyString      STRING('LITERAL')
```

- for numeric data types, a number without quotes generates:

```
MyNumber      LONG(100)
```

Tip

Functions and variables are valid initial values for database columns, but not for global, module, or local memory variables! Use single quotes to specify literal values for database columns, but no quotes for memory variables.

External Name

To specify an external name for the column, type it in the **External Name** field. This covers cases where the column label within the program is different than the name of the column in the data file; for example, you may be accessing a column through an ODBC connection to a database which allows column names longer than the maximum for a Clarion label. Place the name of the column as it exists in the data table here. See also: ODBC Accelerator Driver in the Database Drivers manual.

Initialization

(Application Generator Only) Check the **Do not Initialize** box to place the AUTO attribute on the data element defined. This means that the compiler will not initialize the memory area that is reserved for this data element, and it is the programmer's responsibility to do so.

Place Over

To declare the column as an overlay, select another column name from the drop down list. This allows the current column to redefine the other column's location in memory. The Application Generator adds the OVER attribute.

Storage Class

Check box choices are available here based on the scope of where the variable is defined (global, module, or local). By default, a local variable is allocated from stack memory, which means the variable is reallocated for each new instance of the procedure. By default, global or module data is implicitly static.

STATIC adds the STATIC attribute.

The variable is allocated static memory instead of stack memory, which makes the variable "persistent" from one instance of the procedure to the next.

THREAD adds the THREAD attribute.

The variable is allocated static memory separately for each execution thread in the program. Thus the value of the variable depends on which thread is executing.

Allocation

Check box choices are available here based on the scope of where the variable is defined (EXE or DLL). By default, a variable is allocated memory in the default program.

EXTERNAL adds the EXTERNAL attribute.

Specifies the variable is defined in an external library, and therefore is allocated no memory by this program.

DLL adds EXTERNAL and DLL (dll_mode).

Specifies the variable is defined externally in a DLL. dll_mode is a switch indicating whether the DLL attribute is active or not. The DLL attribute is required for EXTERNAL variables in 32-bit applications.

Comments

Allows you to enter a text description describing the column. The description is solely for your convenience, and has no effect on the application. It is useful for situations in which other programmers may pick up your code later, or for when you expect to return to the project after a long period of time since you last looked at it.

Options

Do Not Auto-Populate This Column

Directs the wizards to skip this column when creating Form, Browse or Report procedures.

Population Order

Specifies the order in which the wizards populate columns. Choose **Normal, First,** or **Last** from the drop down list. Wizards populate in this order: all Columns specified as First, then all Columns specified as Normal, and finally all Columns specified as Last.

Form Tab

Specifies the TAB onto which the wizards populate the column. Type the Caption for the TAB or select one you have previously created from the drop down list. This allows you to direct the wizard to group columns in the manner you want.

Add Extra Vertical Space Before Column Controls on Form Procedures

Check this box to direct the wizards to add vertical space between this column's control and the one populated above it.

User Options

The text typed into this column is available to any Utility Templates that process this file in the %FieldUserOptions symbol. The individual Utility Templates determine the proper syntax for these user options.

This list is divided into two columns, the left entry is for the property label, the right for its value. To add a new entry, RIGHT-CLICK and **choose Insert**. You have four choices: *Bool*, *String, Integer* or *INI*. Bool is for Boolean type expressions, such as On or Off, True or False, etc.

If you choose **String**, the value has an ellipsis button to open a text window. This allows you to enter a string value.

If you select **Integer**, the user-defined option will be a numeric value.

Choose **INI** to get a property that is defined. This is used to set pre-defined properties for use later. A drop list is provided allow you to choose from the list. Selecting a property from this list not only adds the property but also activates it for use. Follow the instructions provided with your add-on template set.

Follow the instructions provided with your add-on template set.

Help

Help ID

To specify a help ID for controls referencing the column, specify a help topic in the **HLP** field. The Application Generator adds the HLP attribute.

Message

To specify a status bar message for controls referencing the column, type the message in the **MSG** field. The Application Generator adds the MSG attribute. When the control referencing the column has focus, the text appears on the status bar, provided the window in which the control appears has one.

Tool Tip

To specify a popup message for controls referencing the column, type the message in the **Tool Tip** field. When the cursor is idle over the control referencing the column, the message appears immediately below the cursor in a popup box. The Application Generator adds the TIP attribute.

Validity Checks

Choose a validation option in this dialog. The Application Generator uses the information when creating and maintaining controls. When the user completes the field and shifts focus to another control, the application will sound a warning beep and set focus back to the control if the data is not valid..

No Checks

Specifies no validation. This is the default.

Note:

Some validation can still occur, depending on the control you use to display the data, and the attributes of the control. For example, an ENTRY control with the REQ attribute automatically enforces a non-blank entry.

Choices Type the choices to display in the format "Choice1|Choice2|Choice3." Separate the choices with a pipe (|) character (usually SHIFT+\). The Application Generator adds the FROM attribute to SPIN, LIST, and COMBO controls, or adds text to RADIO controls (see the *Language Reference*).

Values Type the value to assign when the end user selects the corresponding choice. Type the values in the format "value1|value2|value3." Separate the values with a pipe (|) character (usually SHIFT+\). The Application Generator adds the VALUE attribute to RADIO controls (see the *Language Reference*).

Must be in Numeric Range

Specifies the entry must fall within a numeric range. You may specify a minimum value, a maximum value, or both. The Application Generator generates code to enforce the range you specify, and adds the RANGE attribute to SPIN controls (see the *Language Reference)*.

Lowest Check this box to set a minimum value, then enter the value in the corresponding spin box. Clear the box to specify no minimum value.

Highest Check this box to set a maximum value, then enter the value in the corresponding spin box. Clear the box to specify no maximum value.

By entering only a lowest, or only a highest value, you can specify an open ended range.

Must be True or False

Specifies a Boolean entry (yes/no, true/false, off/on). The Data Dictionary Window Control defaults to CHECK.

True Value Type the value to assign when the end user checks the CHECK control. The Application Generator adds the VALUE attribute to the CHECK control (see the *Language Reference* PDF).

False Value Type the value to assign when the end user clears the CHECK control. The Application Generator adds the VALUE attribute to the CHECK control (see the *Language Reference* PDF).

Note:

The Application Generator does not generate code to enforce true/false entries because, in Clarion, all entries evaluate as either true or false. This selection affects the default window control in the Data Dictionary and applies the VALUE attribute if the control is a CHECK.

Must be in File Specifies the value must match a column in a file. This option is enabled only if you previously related another file or files. See *Adding or Modifying Relationships* in this chapter.

File Label Select the lookup file from the list of related files. The Application Generator generates code to make sure the entered value is in the selected lookup file.

Tip

Use the FileDrop or FileDropCombo control template in your application to provide a same window pick list for the end user, or use the Actions tab for an ENTRY control to provide a separate window pick list.

Must be in List

Specifies the value must match one of the specified choices. The choices are displayed with a SPIN, LIST, COMBO, or RADIO control.

Choices Type the choices to display in the format "Choice1|Choice2|Choice3." Separate the choices with a pipe (|) character (usually SHIFT+\). The Application Generator adds the FROM attribute to SPIN, LIST, and COMBO controls, or adds text to RADIO controls (see the *Language Reference*).

Values Type the value to assign when the end user selects the corresponding choice. Type the values in the format "value1|value2|value3." Separate the values with a pipe (|) character (usually SHIFT+\). The Application Generator adds the VALUE attribute to RADIO controls (see the *Language Reference* PDF).

Note: The Application Generator does not generate code to enforce Must be in List entries. This selection affects the default window control in the Data Dictionary and applies the FROM and VALUE attributes.

Window

To pre-format a window control referencing the current field, select the **Screen Controls** tab, then specify the options in this dialog.

Tip

By choosing the properties for a control at this time, you can save time later. Every application you generate from the dictionary, and every procedure in the application will automatically format the control the way you want it. If you don't format it here, and if the control requires custom formatting, you will have to custom format it for each procedure and application later.

Select either the prompt or entry field from the **Window Controls** list, then press the **Properties** button. The prompt is the label which appears next to the control. The entry field is the actual control which accepts user input.

Window Controls In most cases, this list box will show an ENTRY and PROMPT control for the column. Select the control to preformat.

Properties Allows you to customize the control selected in the **Window Controls** list by displaying its Properties dialog.

Reset Controls Allows you to return the control type to its default, if you changed it by selecting another from the **Control Type** list.

Control Type Allows you to select a different control type consistent with the column type and **Validity Checks** selected. For example, if you chose the **Must be in List** option in that dialog, one of the choices will be a list box.

Depending on whether the control can receive focus, (or in the case of a check box, which places the mnemonic in the label), the PROMPT in the **Window Controls** list is deleted.

An IMAGE control cannot receive focus, and also has no PROMPT.

Report

To pre-format a report control referencing the current column, select the **Report Controls** tab, then specify the options.

Tip

By choosing the properties for a control at this time, you can save time later. Every application you generate from the dictionary, and every procedure in the application will automatically format the control the way you want it. If you don't format it here, and if the control requires custom formatting, you will have to custom format it for each procedure and application later.

Select the string column from the **Report Controls** list, then press the **Properties** button.

Report Controls The STRING control for the column.

Properties Allows you to customize the control selected in the **Report Controls** list by displaying its Properties dialog.

Reset Controls Allows you to return the control type to its default, if you changed it by selecting another from the **Control Type** list.

Control Type Allows you to select a different control type consistent with the column type and **Validity Checks** selected. For example, if you chose the **Must be in List** option in that dialog, one of the choices will be a list box.

Depending on whether the type of control, the first string control in the **Report Controls** list is deleted.

New/Edit Key Properties Dialog

This dialog allows you to define a key for the currently selected file.

The Dictionary Editor allows you to add keys and their components one after another, quickly. Each time you complete and close the Properties dialogs for one key, another blank dialog appears, ready for the next. Press **Cancel** when a blank dialog appears after completing the last key, to return to the **Column/Key Definition** dialog.

General

Key Name *To specify a Clarion label for the key*, type a valid Clarion label in this field.

Tip

Remember that you cannot give a key the same name as one of the fields within the RECORD. One common convention is to use the field name plus the word "key," as in *LastNameKey*.

Description *To place a text description for the key* in the Data Dictionary, type it in this field. The description appears in dialogs such as the *File Definition* dialog. If you anticipate using many keys for your application, we recommend filling in this field.

Tip

Clarion's Wizards use this description when creating Browses, Forms, and Reports.

Type *To specify a record key, static index file, dynamic index file or Order*, choose an option button in the **Type** group. The **Static Index** and **Dynamic Index** options are *disabled* when the **Require Unique Value** box on the Attributes tab is checked, because indexes allow duplicates. **Order** adds a sort order definition to a template symbol (%ORDER) in a similar manner as a KEY without affecting the file declaration.

Note: Orders specified in a dictionary are not yet supported in the included templates. Some third party templates may use this option.

Attributes

External Name *To specify a DOS filename for an external key*, type a valid DOS filename in this field.

Require Unique Value *To disallow multiple records with duplicate values in their keys*, check this box. This option is valid only for keys, and is disabled for indexes.

Primary Key *To establish the current key as the Primary key*, mark this checkbox. The Application Generator adds the PRIMARY attribute. This may be required for certain file drivers.

The primary key must be unique and exclude nulls.

Auto Number *To specify the Application Generator should create code to manage record sequence numbers*, check this box.

Case Sensitive *To sort according to case*, check this box. When creating or updating the key, all capital letters will precede the lower case letters, as per their positions in the ASCII table.

Exclude Empty Keys *To exclude records with a null or zero value from the key*, check this box.

Comments

Allows you to enter a text description describing the key. The description is solely for your convenience, and has no effect on the application. It is useful for situations in which other programmers may pick up your code later, or for when you expect to return to the project after a long period of time since you last looked at it.

Options

Do Not Auto-Populate This Key
Directs the wizards to skip this Key when creating primary Browse procedures or Report procedures.

Population Order Specifies the order in which the wizards populate keys. Choose *Normal, First,* or *Last* from the drop down list. Wizards populate in this order: all Keys specified as First, then all Keys specified as Normal, and finally all Keys specified as Last.

User Options

This list is divided into two columns, the left entry is for the property label, the right for its value. To add a new entry, RIGHT-CLICK and **choose Insert**. You have four choices: *Bool*, *String, Integer* or *INI*. Bool is for Boolean type expressions, such as On or Off, True or False, etc.

If you choose **String**, the value has an ellipsis button to open a text window. This allows you to enter a string value.

If you select **Integer**, the user-defined option will be a numeric value.

Choose **INI** to get a property that is defined in the C60DCTOP.INI file, located in the \BIN folder. This is used to set pre-defined properties for use later in the associated application. A drop list is provided to allow you to choose from the list. Selecting a property from this list not only adds the property but also activates it for use. For add-on INI properties, follow the instructions provided with your add-on template set.

Fields

Specify the components of keys (the field or fields)-using the Fields tab. You may specify more than one field for a key. Each field is appended to the **Keys** list in the **Field/Keys Definition** dialog.

The **Fields** tab features a list displaying the components.

Sort Order

Choose either the **Ascending** or **Descending** radio buttons to specify the order for the highlighted component.

Note: Not all file drivers support mixing ascending and descending components in the same key.

Insert

Calls the Insert Key Component dialog listing the available fields. DOUBLE-CLICK on the name of a field in the list, to place it in the key.

Delete

Removes the highlighted component from the key.

Move up/Move Down buttons

Moves the highlighted component up or down in the list.

New/Edit Relationship Properties Dialog

Set relationships between files in this dialog. The relationships appear in the **Related Files** list on the **Dictionary** dialog, for the currently selected file. When completing this dialog, work from the top down. Start with the **Relationship for** *selected file* group box:

General

Type Set the relationship type by choosing 1:Many or Many:1 from the drop down list.

Key Depending on the relationship type selected, choose a primary or foreign key from the drop down list. The choices in the drop down are the keys previously defined for the currently selected file.

Depending the relationship type you choose for the selected file, the next group box will be labeled either: **Child**, **Parent** or **Link** (respective to 1:Many or Many:1):

Related File Choose another file from the dictionary to relate to the selected file.

Key Depending the relationship type you choose for the selected file, the label for this drop down box will be either Primary or Foreign. Select a previously defined key for the related file from the drop down box.

Field Mapping This group box displays two lists, each showing a key, and the field in the related file which "maps" to it. If the key field names of each file match each other, then just press the **Map by Name** button (below), and Dictionary Editor will automatically link the fields. If they do not, double click on each item in the list boxes, then select a field from the related file that links to the key, in the **Select a Field** dialog.

Map By Name Automatically defines links based on similarly named fields in each data file.

Map By Order Automatically defines links based on the order in which fields are defined in each data file.

Note:

Alternatively, you can map each field manually by double-clicking the field name in the **Field Mapping** list.

After choosing all other options, set the options in the **Referential Integrity Constraints** group box. The Application Generator automatically generates the code that enforces your selections.

Referential Integrity requires that a foreign key cannot contain any value which has no match in the primary key. This raises potential problems when the end user wishes to change or delete the primary key record.

The **On Update** and **On Delete** drop down boxes each offer the following choices:

No Action Instructs the Application Generator *not* to generate any code to maintain referential integrity.

Restrict Instructs the Application Generator to disallow the user from deleting an entry, if the value is used in a foreign key. For example, if the user attempts to change a primary key value, the generated code attempts to check for a related record with the new value, changes it back if necessary, then loops back to the entry dialog so that the user can enter another value.

Cascade Instructs the Application Generator to update or delete the foreign key record. For example, if the user changes a primary key value, the generated code changes the values in the foreign key that referenced the primary key. If the user deletes a primary key value, the code deletes the foreign key value, too.

Clear Instructs the Application Generator to change the value in the foreign key to blank or zero.

Tip

The following server settings have the same effect as the No Action setting (no code is generated); however, they can provide an indication that RI is enforced elsewhere, and their values can be used to trigger template generated scripts for the server.

Restrict (Server)
Tells the Application Generator that the back-end server will prevent the user from deleting or changing an entry if the value is used in a foreign key.

Cascade (Server)

Tells the Application Generator that the back-end server will update or delete the foreign key record.

Clear (Server)

Tells the Application Generator that the back-end server will set the foreign key to null or zero.

Comments

Allows you to enter a text description describing the relationship. The description is solely for your convenience, and has no effect on the application. It is useful for situations in which other programmers may pick up your code later, or for when you expect to return to the project after a long period of time since you last looked at it.

Options

User Options

This list is divided into two columns, the left entry is for the property label, the right for its value. To add a new entry, RIGHT-CLICK and **choose Insert**. You have four choices: *Bool*, *String, Integer* or *INI*. Bool is for Boolean type expressions, such as On or Off, True or False, etc.

If you choose **String**, the value has an ellipsis button to open a text window. This allows you to enter a string value.

If you select **Integer**, the user-defined option will be a numeric value.

Choose **INI** to get a property that is defined. This is used to set pre-defined properties for use later. A drop list is provided allow you to choose from the list. Selecting a property from this list not only adds the property but also activates it for use.

Follow the instructions provided with your add-on template set.

New/Edit Table Properties Dialog

This dialog allows you to add a new data table to the list and choose its database driver.

Once the table appears on the list, you may declare columns, keys, set relationships, and other properties for the data table. Using the data from this dialog, the Application Generator will write the FILE structure declaration.

To modify the table properties at a later time, highlight the table name on the **Dictionary** dialog list, then either DOUBLE-CLICK or press the **Properties** button.

General

Usage

Mark the structure as a Table, Global Data group, or Column Pool. You can define relationships for tables, but not for Global or Pool structures.

Table The structure represents a FILE. The Application Generator generates a FILE declaration as well as code to read and write the FILE. You can use the columns in the FILE as the parents of derived columns. See **Derived From** in the **Column Properties** dialog.

Global

The structure represents a group of global data declarations. The Application Generator generates a global data declaration for each column in the structure. You can use the global columns as the parents of derived columns. See **Derived From** in the **Column Properties** dialog. This option enables the **Generate Last** checkbox which causes these declarations to follow declarations made for global variables defined in the .APP.

Pool The structure represents a Column Pool. The Application Generator generates no code for this structure. You can use the columns in the pool as the parents of derived columns. See **Derived From** in the **Column Properties** dialog.

Name Type a data table name, as you wish to refer to it in your code. This serves as the label for the Clarion FILE structure. Specify a valid Clarion label--Clarion will automatically truncate the name if necessary. You may also specify a completely different name for the DOS file--see **Full Pathname**, below. See also: Declaration and Statement Labels in the Language Reference manual.

Description Enter a string description for the table. Clarion automatically displays the descriptions in certain dialogs, allowing you to quickly recognize the table contents.

Prefix

As you enter the data table **Name**, Clarion automatically extracts the first three letters to use as a label prefix when referring to the table. Optionally specify up to 14 characters of your choice in this field. This supplies the parameter for the PRE attribute.

The prefix allows your application to distinguish between similar variable names occurring in different data structures. A column called *Invoice* may exist in one data table called *Orders* and another called *Sales*. By establishing a unique prefix for *Orders* (ORD) and *Sales* (SAL), the application may refer to columns as ORD:INVOICE and SAL:INVOICE.

Database Driver

Specify the database type. This supplies the first parameter for the DRIVER attribute. When using the Application Generator, Clarion automatically links in the correct database driver library. See *Database Drivers* for a discussion of the relative advantages of each driver. You can specify the default driver by choosing **Setup ▸ Dictionary Options**.

Remember that individual database drivers may vary in their support of some of the attributes which you add to the FILE structure in this dialog box.

Driver Options

Optionally type a string for the second parameter of the DRIVER attribute. This conveys additional instructions to the database driver and corresponds to the second parameter for the DRIVER attribute, also known as a "driver string." *Database Drivers* contains additional information. This supplies the second parameter for the DRIVER attribute.

Owner Name

Optionally type a string containing the password for access to the table. This is dependent on the database system. This adds the OWNER attribute to the FILE statement.

You may also check the **Encrypt** box (below) which adds the ENCRYPT attribute. Encrypting the table means that only your application will be able to read the table. It does not mean that it automatically prompts the end user for a password. The end user, however, may not access the data with any other table viewer.

When using the ODBC driver, type the data source name, user ID, and password, separated by commas, in this field.

Full Pathname

Type either the path, or a fully qualified file name for the data table. If you leave the file name out, Clarion automatically uses the first eight letters of the name entered in the **Name** field. You may also omit the table extension--Clarion will supply the correct extension depending on the database driver chosen. This supplies the parameter for the NAME attribute.

When using the TopSpeed driver, if you wish to store multiple tables in a single physical file, separate the file and table names with "`\!`," as in **`TUTORIAL\!ORDERS.`** This refers to the ORDERS table in the TUTORIAL.TPS file.

When using an ODBC driver to define a FILE such as Microsoft Access, which can store multiple tables in a single file, place the table name in this field. Typically, the name of the physical file that includes the table is listed in the ODBC.INI file; the ODBC driver manager provides this information to the driver.

Tip

To specify a variable name for the actual file name, place it in this field, and prefix the variable name with an exclamation point (!).

Enable Table Creation

Optionally specify that the application should create the data table if it does not exist at runtime. This adds the CREATE attribute to the FILE statement.

Reclaim Deleted Records

This option is dependent upon the database driver. It specifies that the application reuse file space formerly taken up by deleted records. Otherwise, the application adds new records to the end of the table. This adds the RECLAIM attribute to the FILE statement.

Encrypt Data Records

Optionally turn on table encryption. You must also specify an **Owner Name** (see above). This adds the ENCRYPT attribute to the FILE statement.

Open in Current Thread

Optionally specify that each execution thread in your application that uses this table allocates memory for its own separate record buffer. This is typically for use in multiple document applications, and improves table handling. The Clarion default templates automatically add the THREAD attribute on each FILE structure.

Use OEM Collation
Specifies string data is converted from OEM ASCII to ANSI when read from disk and ANSI to OEM ASCII before writing to disk. This adds the OEM attribute to the table definition.

Enable Column Binding
Optionally specify that all variables in the RECORD structure are available for use in dynamic expressions at runtime. The compiler will allocate memory to hold the full Prefix:Name for each variable, instead of using its own internal reference for each variable. Therefore the BINDABLE attribute increases the amount of memory necessary for the application.

Export Columns
Check this box to indicate that the columns should be exported from DLLs (valid for Global tables only).

32 Bit Only Optionally specify that the FILE structure is only valid for 32-bit applications (SQL Drivers only).

Freeze Check this box to prevent any derived columns in the table from refreshing. See **Derived From** in the **Column Properties** dialog. See also the **Refresh Column, Refresh Table, Refresh Dictionary,** and **Distribute Column** menu commands.

Comments

Allows you to enter a text description describing the table. The description is solely for your convenience, and has no effect on the application. It is useful for situations in which other programmers may pick up your code later, or for when you expect to return to the project after a long period of time since you last looked at it.

Options

Do Not Auto-Populate This Table

Directs the wizards to skip this table when creating primary Browse procedures or Report procedures.

User Options The text typed into this field is available to any Utility Templates that process this table in the %FileUserOptions symbol. The individual Utility Templates determine the proper syntax for these user options.

This list is divided into two columns, the left entry is for the property label, the right for its value. To add a new entry, RIGHT-CLICK and **choose Insert**. You have four choices: *Bool*, *String, Integer* or *INI*. Bool is for Boolean type expressions, such as On or Off, True or False, etc.

If you choose **String**, the value has an ellipsis button to open a text window. This allows you to enter a string value.

If you select **Integer**, the user-defined option will be a numeric value.

Choose **INI** to get a property that is defined. This is used to set pre-defined properties for use later. A drop list is provided allow you to choose from the list. Selecting a property from this list not only adds the property but also activates it for use

Follow the instructions provided with your add-on template set.

New Table Alias Dialog

An alias creates a second reference for a table without duplicating the file on disk. You can add an alias for a table only if it is already on the Dictionary list.

General

Name Type a data table "name", as you wish to refer to it in your code. The name must be a valid Clarion label.

Description Enter a string description for the alias. Clarion displays the descriptions in dialogs such as the **Dictionary** dialog.

Prefix By default, Clarion will use the first three letters of the Name for the prefix. Optionally specify up to 14 characters to use as a Prefix.

Alias Table Choose a table from the drop down list. This is the *original* table that the alias "references." The drop down list shows only the tables previously defined using the **Add Table** command in the **Dictionary Properties** dialog.

Comments

Allows you to enter a text description describing the alias. The description is solely for your convenience, and has no effect on the application. It is useful for situations in which other programmers may pick up your code later, or for when you expect to return to the project after a long period of time since you last looked at it.

Options

Do Not Auto-Populate This Aliased Table
Directs the wizards to skip the Aliased Table when creating primary Browse procedures or Report procedures.

User Options User Options are provided to enable you to provide information to be used by a third-party template set. User Options are comma delimited, that is, each entry is separated by a comma.

Follow the instructions provided with your add-on template set.

A table alias provides several advantages, at the cost of some system overhead:

- *It allows you to set multiple relationships between tables*.

  Strict relational database theoreticians state a table may only have a single relational link to another table at a time. Aliases allow you to "legally" work around this limitation. See also: How to Design Your Dictionary and Database in the Online User's Guide.

- *It allows a second table buffer for the same table*.

  You could use this for a second table browse, as well as entry forms and other items for each. This is particularly useful for an MDI application.

- *On the negative side, the second table buffer takes up additional memory and resources*.

  Any table driver utilizing external key files requires additional file handles for each alias. For example, a table with three external keys and three aliases requires sixteen file handles: one each for the "first" data table and its three keys, and an additional four for each of the aliases. When using aliases, we recommend choosing a database driver that stores keys internally, such as TopSpeed or Btrieve.

- *When using aliases, you must open the table in Share mode*.

To modify the alias properties at a later time, highlight the table name on the **Dictionary** dialog list, then either double-click or press the **Properties** button.

You can edit the columns and keys for the Alias by pressing the **Columns/Keys** button. The Columns/Key Definition dialog lists the columns and keys for the *original* table; any changes you make will update the originals.

Password Validation Dialog

Allows you to password protect your dictionary, to prevent other developers from modifying it.

To add a password to the data dictionary:

1. Press the **Password** button.

2. When the **Password Validation** dialog appears, type a password in the space provided.

3. Press the **OK** button.

See the *Development and Deployment Strategies* appendix in the Online *User's Guide* for more information.

Revert Dictionary Version

Version ▸ Revert

Rolls back changes in the data dictionary to the last checkpoint.

The Dictionary Editor automatically places an internal version number in your dictionary file. A new dictionary begins with version 1.0. You can see the version number/revision number on the caption bar of the **Dictionary** dialog. The **Dictionary Properties** dialog also displays the original creation data and time, and the last modified date and time.

You should increase the version number, manually, whenever you make significant changes to a dictionary; for example, when you're working on version #2 of your application, choose **Version ▸ Checkpoint**. The revision number (r. #) on the caption bar increases by one.

To roll back to a previous version, choose **Version ▸ Revert**. Choose the revision to revert to by selecting it with the spin control in the **Previous Revision** dialog.

Revert to Previous Revision

This window allows you to rollback a dictionary to a previous version.

Current	Displays the current version.
New Revision	Allows you to select the version to which the dictionary will revert.

Select a Column Dialog

Allows you to select a column (from the related table) to link in the **Field Mapping** lists in the **New Relationship Properties** dialogs. Open the dialog by DOUBLE-CLICKING an item in either list.

Columns list	DOUBLE-CLICK or select a column from the related table, and press the **Select** button.
No Link	Allows you to indicate that a column is not part of the relationship.

Select Column Window

The Select Column window presents a relation tree of tables defined in the dictionary. Press the Expand (+) button to display columns attached to a given file, and press the select button to copy the field name to the appropriate entry.

Select Field Dimension Dialog

This dialog allows you to specify the component of a dimensioned variable which you wish to "attach" to a control, via the **Table Schematic Definition** dialog.

The number of dimensions you specified in the **Field Properties** dialog when you defined the variable determines how many spin boxes are enabled. Choose the specific element by setting the spin boxes.

For example, to refer to the variable DimVariable[4,5], set the first spin box to *4*, and the second to *5*.

Trigger Properties

This dialog allows you to enter a trigger for the selected dictionary table. The following prompts are provided:

General

Type

Select the trigger type to designate when the trigger will be executed during one of the following database actions:

Before Insert	After Insert
Before Update	After Update
Before Delete	After Delete

Data

Press the Data button to add variables that will be used in your trigger. These data elements will automatically be included in applications that reference the active table.

Text

Enter the trigger statement in the list box provided by pressing the ellipsis button to enter a full screen editor format.

Note:

Make sure to press the **OK** button to save your dictionary trigger after returning from the **Data** or **Text** areas, before attempting to add an additional one.

Comments

Allows you to enter a text description describing the trigger. The description is solely for your convenience, and has no effect on the application. It is useful for situations in which other programmers may pick up your code later, or for when you expect to return to the project after a long period of time since you last looked at it.

Driver String Dialogs

Driver String Builder Dialogs

The Dictionary Editor's Table Properties dialog now includes an ellipsis button next to the Driver String prompt that guides you through the valid options of the selected file format.

The Driver String dialog that will be displayed is dependent on the current File Driver selected. For example, the TopSpeed driver contains only two possible driver string settings, where the Btrieve or MS-SQL dialogs can contain many selections on multiple tab controls.

Note:

For more information regarding the particular driver strings, please refer to the *Database Drivers* PDF, and the online help.

Synchronizer Wizard

Overview

The Dictionary Synchronizer synchronizes the open Clarion data dictionary with another Clarion data dictionary (*.DCT), a Clarion text dictionary (*.TXD), or a non-Clarion database such as ORACLE, MSSQL, or Btrieve.

What the Synchronizer Does

Generally

Generally speaking, the Dictionary Synchronizer

- automates the conversion of existing data between different versions of your software;
- automates the synchronization, in either direction, of the "master" data dictionary or database with other project data dictionaries and databases;
- creates a complete Clarion data dictionary, including relationships, from an existing database (SQL, Btrieve, etc.) in a single pass.

Specifically

More specifically, the Dictionary Synchronizer

- compares a Clarion data dictionary with another Clarion data dictionary or with a database (SQL or Btrieve) to identify any differences between them,
- resolves the differences (with your interactive input, or according to a prior synchronization) by proposing changes to the target dictionary or database,
- validates the proposed changes,
- backs up the target dictionary or (non-SQL) database,
- implements the proposed changes directly to the target dictionary (.dct, .txd, or .ddf), or generates an SQL script to implement the changes to the SQL database,
- generates an SQL script or Clarion program to implement the proposed changes to any existing data,
- saves pertinent information about the process for subsequent reuse.

This functionality aids team development by automating the synchronization, in either direction, of the "master" dictionary or database with other project dictionaries and databases. It lets you easily convert or upgrade existing data to higher or later versions of your software, which is a major benefit to both team and individual developers. Finally, the power of Dictionary Synchronizer applies not only to Clarion data dictionaries, but also to non-Clarion file systems and databases such as Oracle, MSSQL, Scalable SQL, Btrieve, etc., so you can create complete Clarion data dictionaries froma existing databases in a single pass.

Synchronizer Servers

The Dictionary Synchronizer uses Synchronizer Servers to access data dictionaries and databases. A Synchronizer Server is a .dll that communicates with a database or file system such as ORACLE, MSSQL, Btrieve, etc.

Generally speaking, the Synchronizer Server does two things:

- During the dictionary comparison stage, it queries the dictionary or database and collects a complete, current description of the database's files or tables, keys, views, relationships, objects and properties.
- During the implementation stage, it generates and optionally executes the programs and database commands necessary to carry out any proposed changes, including converting existing SQL data.

There is a built in Synchronizer Server for Clarion data dictionaries, plus several separate Synchronizer Servers for SQL and Btrieve databases. The Clarion Dictionary server handles all Clarion data dictionaries (regardless of the file driver), and need not be registered.

To register a Synchronizer Server, simply register the corresponding file driver with the Database Driver Registry. See *Clarion's Development Environment—Database Driver Registry* in the *User's Guide* for more information.

Running the Synchronizer

To run the Dictionary Synchronizer, first, be sure to register the appropriate database driver (see *Synchronizer Servers* for more information). Next, open your Clarion data dictionary, then press the **Synchronize** button. This starts the Synchronizer Wizard which leads you step-by-step through the synchronization process. The process requires that you

- identify the *other* dictionary or database with which to synchronize,
- specify the *direction* of synchronization (identify source and target),
- tell the synchronizer *what* to synchronize (all files or some subset of files),
- tell the synchronizer *how to match* files, fields, keys, and relationships (by name, by order, manually, or the same as the last time you synchronized these two dictionaries),
- tell the synchronizer *how to resolve differences* between the dictionaries (copy, delete, ignore, etc.).

Note: The Dictionary Synchronizer does not backup data. We strongly recommend backing up your data prior to any conversion or synchronization.

Synchronizer Wizard - Synchronize With

The Dictionary Synchronizer uses Synchronizer Servers to access data dictionaries and databases. A Synchronizer Server is a DLL that communicates with a database or file system such as ORACLE, MSSQL, Btrieve, etc.

There is a built in Synchronizer Server for Clarion data dictionaries, plus several separate Synchronizer Servers for SQL and Btrieve databases. The Clarion Dictionary server handles all Clarion data dictionaries (regardless of the file driver), and need not be registered.

To register a Synchronizer Server, simply register the corresponding file driver with the Database Driver Registry. See *Clarion's Development Environment—Database Driver Registry* in the *User's Guide* for more information.

This dialog is also used to start the Multi Table import option from your data dictionary

CLICK on the **Next** button to continue.

Synchronizer Wizard - Select Synchronize Direction

The Data Dictionary Synchronizer never modifies the source dictionary; it only modifies the target dictionary.

Specify whether the open data dictionary is the source or the target by CLICKING the appropriate radio button.

Note: Whenever you consolidate two Clarion dictionaries into a single dictionary, you risk damaging the relationship between the replaced dictionary and its corresponding applications, unless you first export those applications to text (.TXA) format. Using Version Control accomplishes this without the need of explicitly creating a .TXA file, but the tutorial in *Enterprise Tools* shows you how to accomplish the same purpose without Version Control.

CLICK on the **Next** button to continue.

Synchronizer Wizard - Select files from Clarion Dictionary

Select which files, views, and tables within the Clarion data dictionary to include in the synchronization process. The Synchronizer Wizard provides two file lists for the data dictionary: on the left is a list of files *not* synchronized and on the right is a list of files *to* synchronize. Depending on the database server, the list may show filenames, table names, owner names, etc.

Tip

The Synchronizer Wizard lists the files in the order specified on the Options page, that is, alphabetically or according to the source dictionary.

It is to your advantage to include as few files as possible in the synchronization process because fewer files results in a faster process.

Add >

Press this button to move the highlighted file *and all related files and aliases* to the **Files to Synchronize** list. Alternatively, DOUBLE-CLICK the file to synchronize.

< Remove

Press this button to move the highlighted file *and all related files* to the **Files not Synchronized** list.

Add All >>

Press this button to move *all* the files to the **Files to Synchronize** list.

<< Remove All

Press this button to move *all* the files to the **Files not Synchronized** list.

CLICK on the **Next** button to select which files within the other database/dictionary to include in the synchronization process. This page works exactly like the previous file selection page, with one exception: if the other dictionary is a not a Clarion dictionary, loading and updating the file list takes significantly longer than for a Clarion dictionary.

CLICK on the **Finish** button to continue.

Synchronizer Wizard - Select files from other Dictionary

Select which files, views, and tables within the Clarion data dictionary to include in the synchronization process. The Synchronizer Wizard provides two file lists for the data dictionary: on the left is a list of files *not* synchronized and on the right is a list of files *to* synchronize. Depending on the database server, the list may show filenames, table names, owner names, etc.

Tip

The Synchronizer Wizard lists the files in the order specified on the Options page, that is, alphabetically or according to the source dictionary.

It is to your advantage to include as few files as possible in the synchronization process because fewer files results in a faster process.

Add > Press this button to move the highlighted file *and all related files and aliases* to the **Files to Synchronize** list. Alternatively, DOUBLE-CLICK the file to synchronize.

< Remove Press this button to move the highlighted file *and all related files* to the **Files not Synchronized** list.

Add All >> Press this button to move *all* the files to the **Files to Synchronize** list.

<< Remove All Press this button to move *all* the files to the **Files not Synchronized** list.

Clustering Clustering specified that you control how many related tables you wish to import.

- *None* — No clustering, just import the highlighted table.
- *Partial* — Some clustering, import the highlighted table and its child tables only.
- *Full* — All related tables. Imports the highlighted table and all its child tables, plus their child tables and so on.

CLICK on the **Next** button to select which files within the other database/dictionary to include in the synchronization process. This page works exactly like the previous file selection page, with one exception: if the other dictionary is a not a Clarion dictionary, loading and updating the file list takes significantly longer than for a Clarion dictionary.

CLICK on the **Finish** button to continue.

Synchronizer Main Window

When you "finish" the **Synchronizer Wizard**, it opens the **Synchronize Dictionaries** dialog.

The **Synchronize Dictionaries** dialog compares the two data dictionaries, item by item and lets you resolve any differences between them. It validates the proposed changes, then generates all the source code, scripts, and data definitions required to implement the changes. Finally, it gives you the opportunity to review and edit the generated source code or SQL scripts.

Overview

The **Synchronize Dictionaries** dialog uses a "file centric" hierarchical list to present the comparison. That is, beneath each file in the list, the **Synchronize Dictionaries** dialog nests the file's properties and components (fields, keys, relationships, and aliases). Beneath each item in the list, the **Synchronize Dictionaries** dialog nests that item's respective properties and components.

Tip: CLICK on the plus (+) sign to expand a list item; CLICK on the minus (-) sign to contract an item.

For every file, field, key, relationship, alias, and property in the list, the **Synchronize Dictionaries** dialog shows the value from each database or dictionary, plus status indicators to show whether the values are different, and how the difference is resolved (for example, ignore the difference, copy from source to target, delete item from target, not yet resolved, etc.).

The **Synchronize Dictionaries** dialog either dims or omits database properties the target database/dictionary does not support, depending on the settings in the **Synchronizer Options** dialog (see *Dictionary Synchronizer Options*).

Generally, follow these steps to synchronize the dictionaries:

1. Configure the **Synchronize Dictionaries** dialog to sort dictionary items according to your preferences—alphabetically or according to the source dictionary (use the **View** menu).
2. Navigate to each unmatched item and make sure each one is properly matched to the corresponding item in the other dictionary by selecting a matching option from the **Synchronize** menu or the popup (RIGHT-CLICK) menu. Items may be left unmatched if there is no corresponding item in the other dictionary.
3. Navigate to each difference and resolve it by selecting an option from the **Synchronize** menu or the popup (RIGHT-CLICK) menu.
4. Press the **OK** button.

Configuring the Synchronize Dictionaries Dialog

By default, the **Synchronize Dictionaries** dialog displays files, fields, and keys in the order specified in the **Synchronizer Options** dialog. You can also set default colors and other behaviors with this dialog (see *Dictionary Synchronizer Options*). However, you can reset the various sort orders (files, fields, keys) at any time with the **View** menu.

Status Indicators

The **Synchronize Dictionaries** dialog uses status indicators as well as color to provide information about the current synchronization process.

Status Indicators

Status indicators are in the right-most column of the list. Status indicators include:

- File Structure (field order) changed. This indicates a field's *position* has changed within the target dictionary. This indicator is especially useful when fields are sorted alphabetically and their physical order is not readily visible.

  Invalid proposal below. This indicates there is an invalid proposal further down in the hierarchy. Expand the tree or use the navigation controls to find the violation.

  Invalid proposal. This indicates the proposed change cannot be accepted by the target server and disables the OK button. For example, number of characters not valid for current data type.

  RIGHT-CLICK on this icon then choose Display Validation Error to see an explanation of the violation.

Note: If you have elected to resolve conflicts automatically (see *Dictionary Synchronizer Options*) and the conflict can be resolved automatically, then this status indicator is not set!

Color

Color indicators include:

Blue	indicates no action is applied, and the item is either different than its matching item or it has no matching item.
Black	indicates an action is applied and the resulting target item is valid, or no action is applied but none is needed because the items match and the target item is valid.
Red	indicates an action is applied and the resulting target item is not valid—the target does not support the applied property. You must resolve the invalid proposal before you can complete the synchronization.
Gray	The item is not supported by the target dictionary.

Navigating the Synchronizer Tree

The **Synchronizer** provides the following navigation aids to let you quickly identify and resolve differences between the two data dictionaries:

Go to the previous difference—automatically expands the list as required. Alternatively choose **Edit ▸ Previous Difference**.

Go to the next difference—automatically expands the list as required. Alternatively choose Edit ▸ Next Difference.

Go to the previous invalid proposal—automatically expands the list as required. Alternatively choose Edit ▸ Previous Invalid Proposal.

Go to the next invalid proposal—automatically expands the list as required. Alternatively choose Edit ▸ Next Invalid Proposal.

Matching Unmatched Items

The **Synchronize** menu (and the RIGHT-CLICK popup menu) contains all the functions needed to match items between the two dictionaries.

Match	Associates the selected item with an unmatched item in the other dictionary. The Synchronizer moves the matched items onto the same row. RIGHT-CLICK or press ESC to abort the match.
Unmatch	This uncouples the selected items and moves them to separate rows so they can be rematched. This is for items that were matched inadvertently or which should not be matched.
Match by Name	Unmatches the selected item and rematches it by name.
Match by Order	Unmatches the selected item and rematches it by order of the source dictionary.
Match by Component	Unmatches the selected key and rematches it based on component fields.

Tip

DRAG from a source column in one row and DROP on a target column in another row to match the two items. If either item was already matched, this operation first unmatches the original match.

Relationships are matched automatically, based on the files and keys the relationship uses.

Resolving Differences Between Dictionaries

All differences between source and target dictionaries can be resolved by applying an appropriate operations to each item. The **Synchronize** menu (and the RIGHT-CLICK popup menu) contains all the operations needed to resolve differences between the two dictionaries.

The **Synchronize Dictionaries** dialog uses action indicators to provide information about the current synchronization process.

Synchronizer Actions and Action Icons

Action icons are between the source and target columns of the list. Applying an action to any item in the list affects the selected item *and its children*. The action applied to an item cascades to any nested items. The primary item's action icon is in full color; the cascaded action indicators are gray.

Tip

Choose Edit ▸ Undo to undo the last action.

The **Synchronize Dictionaries** dialog shows the proposed action for each item in the list. The action icons are in the column between the two dictionaries. Synchronizer actions and their corresponding action icons are:

Icon Menu Selection (and explanation of synchronizer action).

	No decision. This is not a menu selection. It is the initial state of the **Synchronize Dictionaries** dialog before any actions are applied. This state does not change the target dictionary.
Add to target	Add only. Adds the selected source dictionary items to the target dictionary. This applies only to source items with no matching target items. This action only adds new items in the target dictionary.
Copy to target	Change only. Copies the properties of the selected items from the source to the target. This action only affects matched items—it ignores unmatched items. This action only changes existing items in the target dictionary.
Merge into target	Add and change. For matching items, this action copies properties from source to target; it adds unmatched source items to the target dictionary; it ignores any unmatched target items. This action can add new items to, and change existing items in the target dictionary. This action makes the target dictionary a union of the source and target.
Replace target	Add, change, and delete. For matching items, this action copies properties from source to target; it adds unmatched source items to the target dictionary; it *deletes unmatched target items***.** This action can add new items to, change existing items in, and delete items from the target dictionary. This action changes the target item to exactly match the source item.

Delete from target	Delete only. Deletes the selected items from the target dictionary. This applies only to target items with no matching source items.
Ignore Difference	Ignore the selected items. This action does not change the target dictionary.
Set to Default	Parent. Applies the parent's action to the selected items. This action is available only if you have previously changed a child's default operation.

Tip

Choose Edit ▸ Undo to undo the last menu selection.

Implementing the Changes

When you have matched all relevant database items, and resolved all differences and invalid proposals in the **Synchronize Dictionaries** dialog, you can implement any proposed changes by pressing the **OK** button.

The **Synchronizer** creates the data conversion program, backs up the target dictionary, and generates the new target dictionary or database definition in the appropriate format—Clarion .DCT or .TXD file, Btrieve .DDF files, or SQL script.

You may execute the SQL script, or you may save the script, optionally edit it, and execute it later. The SQL script both updates the structure of the database and converts existing data to the new structure.

The **Synchronizer** creates back up files by renaming the original target file—**Synchronizer** changes the first position of the file extension to a 'B.' So *.DCT is backed up to *.BCT, *.TXD is backed up to *.BXD, *.DDF is backed up to *.BDF, etc.

Running the SQL Script

How you run the generated SQL script depends on the SQL server. See your SQL database documentation for instructions on storing and running scripts.

Synchronizer Options

Sort files alphabetically

Checking this box displays files and aliases in alphabetical order on the **Synchronize Dictionaries** dialog. Clearing the box displays them in the order they are defined in the source database.

You may also specify the display order during synchronization by choosing **View ▸ Sort files alphabetically**.

Sort fields alphabetically

Checking this box displays fields in alphabetical order on the **Synchronize Dictionaries** dialog. Clearing the box displays them in the order they are defined in the source database.

You may also specify the display order during synchronization by choosing **View ▸ Sort fields alphabetically**.

Sort keys alphabetically

Checking this box displays keys in alphabetical order on the **Synchronize Dictionaries** dialog. Clearing the box displays them in the order they are defined in the source database.

You may also specify the display order during synchronization by choosing **View ▸ Sort keys alphabetically**.

Create conversion program

Checking this box tells the synchronizer to automatically generate a Clarion program to convert existing data to the new format of the target database.

Note: This box is disabled when the target is an SQL database, because no separate conversion program is required. Rather, the synchronizer generates an SQL script to both change the database definition and convert existing data.

Hide unsupported attributes

Checking this box tells the synchronizer to hide (omitted from the **Synchronize Dictionaries** dialog) any attributes of the database the synchronizer server does not support. Clearing the box tells the synchronizer to display the unsupported attributes, but as disabled (dimmed) items.

Correct validity violations automatically

Checking this box tells the synchronizer to automatically resolve conflicts where possible. Clearing the box tells the synchronizer to display the Invalid Proposal icon so you can resolve the problem manually.

For example, a field data type is record picture, but the target dictionary does not support record picture, so the synchronizer automatically resolves the violation by applying the same operation on record picture as on data type.

Note:

The synchronizer cannot automatically resolve some conflicts, such as maximum number of MEMO fields exceeded.

If you have elected to resolve conflicts automatically and the conflict can be resolved automatically, then the unresolved difference status indicator in the Synchronize Dictionaries dialog is not set!

Colors

The **Synchronize Dictionaries** dialog uses color to provide information about the current synchronization process. You can use the default colors or specify your own colors.

Select a synchronizer element in the **Elements** list box, then CLICK on a color selection box. The sample text shows you how the selected element will appear in the **Synchronize Dictionaries** dialog.

Synchronizer Wizard - Select Other Dictionary

The Data Dictionary Synchronizer synchronizes the active dictionary in the Clarion Dictionary Editor with another database or data dictionary.

Press the ellipsis button to select the "other" database or dictionary with the Windows file dialog. If the data dictionary resides in several files (such as a set of Btrieve .DDFs), then select the directory that contains the definition files.

For SQL databases (such as ORACLE), the Synchronizer Wizard prompts you for the logon information needed to connect to the SQL database.

CLICK on the **Next** button to continue.

Synchronizer Wizard - Server Login Info

Provide the appropriate connection information for your backend database. This is specific to the type of backend database you are using.

Synchronizer Wizard - Select Previous Session File

The Previous Session file (*.SYN) stores information about the synchronization of two particular dictionaries. Specifically, it stores the files to synchronize from the source dictionary, the files to synchronize from the target dictionary, and a list of matching items (files, fields, keys, and aliases) between the two dictionaries.

If you wish to save the synchronization information, or if you wish to reuse the information from a prior synchronization, you may specify a filename for the Previous Session file.

Press the ellipsis button to select a Previous Session file with the Windows file dialog, or type the Previous Session file pathname.

CLICK on the **Next** button to continue.

Synchronizer Wizard - Options

These wizard synchronizer options operate exactly the same as the global synchronizer options described in the *Dictionary Synchronizer Options* section. Changing the settings here automatically updates the global settings and vice versa. See the *Dictionary Synchronizer Options* section for more information.

CLICK on the **Matchings** button to tell the synchronizer how to match files, fields, keys, and aliases between the two dictionaries.

Tip: Proper matching of dictionary items is essential to producing a useful synchronization.

Synchronizer Wizard - Matching Options

Proper matching of dictionary items is essential to produce a useful synchronization. The synchronizer provides a range of automatic, intelligent matching choices, which you can combine with manual matching to quickly and accurately match your dictionary items.

Any items the synchronizer cannot match automatically are flagged within the **Synchronize Dictionaries** dialog so you can manually match them. See *Synchronize Dictionaries Dialog* for more information.

Historically

The **Synchronizer** matches the items based on the synchronizer file you specified earlier. This choice is disabled if you specified a new synchronizer file or no synchronizer file.

By Name Only

The **Synchronizer** matches the items based on their labels and external names.

By Order Only

The **Synchronizer** matches the items based on the order they appear in the two dictionaries.

This option is not available for SQL databases.

By Component Only

The synchronizer matches keys based on the number and labels or external names of the key components.

This option is not available for SQL databases.

Note:

The availability of the matching rules depends on the particular database server.

Manually

The **Synchronizer** does not match items. You must match the items with the **Synchronize D**ictionaries dialog.

CLICK on the **OK** button to return to the Options page, then CLICK on the **Next** button to continue.

Note:

Pressing the finish button at this point matches the dictionaries according to these settings. For any manually matched items, all sub-items are matched using the settings. For example, if you match a file manually, all its fields are matched by name if By Name Only is specified.

Synchronizer Validation Error Window

This window provides an explanation of the validation error. This error must be resolved before you can synchronize the dictionaries.

Synchronizer Options Dialog

The Dictionary Synchronizer is configurable. That is, to some extent you can determine how the synchronizer looks and acts. The main synchronizer elements you can configure are

- colors,
- sort sequences,
- automatic generation of data conversion program,
- automatic resolution of conflicts.

The default settings are suitable for many cases so you probably don't need to change these settings at first. However, at some point you may want to change the default settings.

- To reconfigure the synchronizer, choose **Setup ▸ Dictionary Synchronizer Options**.

Tip: You can change many of these settings with the View menu while the Synchronize Dictionaries dialog is open.

Batch Synchronization

For every synchronization, the Dictionary Synchronizer creates a log called synlog.txt in the root directory of the current drive (root:\synlog.txt). This log serves two purposes: it can be used to rerun a synchronization in batch mode, and it should be submitted with any bug reports along with the synchronized data dictionaries to aid in the debugging process.

After you finish a synchronization (either successfully or unsuccessfully), you can use this log file to automatically rerun the synchronization in batch mode. This can save lots of time for large or repetitive synchronization tasks.

To Synchronize in batch mode:

1. Copy or rename the root:\synlog.txt to another location (otherwise the next synchronization will overwrite it).
2. Use the copy of synlog.txt as the first parameter in the RerunSynchronizer DDE command (see example program).

   The second parameter is the Clarion dictionary that started the original synchronization—not necessarily the source dictionary.
3. Run the example program.

   The synchronizer starts and repeats the steps logged in the synlog.txt file.

Note:

If you pressed OK or Cancel at the end of a synchronize, you may want to delete those commands from the synlog.txt file.

```
BatchSynchronize   PROGRAM

DDEChannel      LONG
DDEErrorMsg     CSTRING(300)
DDEActiveDct    CSTRING(300)
DDEErrorNum     USHORT
BatchLog        STRING('c:\SyncTutr\batchlog.txt')
ClarionDct      STRING('c:\SyncTutr\tutorial.dct')

  MAP
    INCLUDE('DDE.CLW')
  END
  CODE
  System{PROP:DDETimeOut} = 12000       !Time out after two minutes
  DDEChannel = DDEClient('ClarionWin')
  IF DDEChannel < 1
    RUN('CW')
    LOOP
      DDEChannel = DDEClient('ClarionWin')
      IF DDEChannel < 1 THEN CYCLE.
    END
  END
  DDEExecute(DDEChannel,'[RerunSynchronizer('&BatchLog&','&ClarionDct&')]')
  DO CheckDDEError
  DDEClose(DDEChannel)

CheckDDEError   ROUTINE
  DDEErrorMsg = ''
  err# = ERRORCODE()
  IF err# > 600
    ! DDEExecute Failed
    IF err# = 603
      DDERead(DDEChannel, DDE:manual, 'GetErrorNum', DDEErrorNum)
      DDERead(DDEChannel, DDE:manual, 'GetErrorMsg', DDEErrorMsg)
      MESSAGE('Error ' & DDEErrorNum & ' : ' & CLIP(DDEErrorMsg))
    ELSIF err# = 605
      MESSAGE('DDE timeout')
    ELSE
      MESSAGE(err#)
    END
  END
```

File Import Window

The File Import dialog is used to enter the required connection information to a data source that you are attempting to import into your data dictionary.

Connection
Enter or paste a valid connection string to use to connect to your data source. Press the ellipsis button to call the Microsoft Data Link interface, which is used to construct a valid connection string for you.

User ID
If your data source requires a User ID to connect, enter it here, or select a previously used User ID from the drop list provided.

Password
If your data source requires a valid password to connect, enter it here, or select a previously used Password from the drop list provided.

Include/Exclude System Files
Check the appropriate radio button to include or exclude all system files associated with the data source. Normally, system files are administrative in nature and should be excluded from your import.

Select Table to Import
With a single table import, you will be presented with a list box of valid import tables to choose from. Highlight the table you wish to import, and press the Finish button to complete the import process

Resolving Invalid Proposals

Invalid proposed changes to the target dictionary can be resolved in two ways: from within the Synchronizer and from outside the synchronizer.

Within the Synchronizer, choose **Display Validation Error** from the menu to find out exactly what the problem is. **Display Validation Error**
Provides information about the nature of the invalid proposal—describes why it is invalid.

If you don't need the offending item in the target dictionary/database, you can proceed with the synchronization by unmatching the item (RIGHT-CLICK the item then choose **UnMatch** from the popup menu), then deleting it from the target (RIGHT-CLICK the item then choose **Delete** from the popup menu).

If you do need the item in the target, you may be able to resolve the violation by changing the Synchronizer operation applied to the item. For example, if you copied the item from the source to the target (you chose **Copy to Target** from the menu), then all properties of the source item (including any unsupported properties) are applied to the target item, resulting in an invalid proposal. Changing the operation from "copy" to "ignore" should resolve the problem (RIGHT-CLICK the item then choose **Ignore Differences** from the popup menu) because the Synchronizer does not copy the source item properties to the target item.

Alternatively, you may resolve the violation by canceling the synchronizer session, changing the offending data declaration in the source dictionary/database, then resynchronizing.

Project System

Database Driver Libraries

Your application calls various database driver routines to access your database tables. These routines are in libraries supplied with Clarion and installed by default in the \LIB subdirectory by the Clarion setup program. During the *link* process, references to these external routines can only be resolved if the library containing the routines is added to your project file.

Note:

The Application Generator automatically adds the appropriate driver libraries based on the Data Dictionary table driver selections and the Project System settings. For hand-coded projects, you should manually add the appropriate driver libraries.

To link a database driver library, **highlight** *Database driver libraries* in the Project Tree list. Then **press** the **Add File** button and select the driver to add from the **Select Driver** dialog.

New Project File Dialog

This dialog allows you to type in the basic information the Project System needs to create a new project file (.PRJ) for you.

Project Title
Allows you to type in a short text description which displays at the top of the Tree List in the Project Tree dialog.

Main File
Type in (or select with the Open File dialog after pressing the ellipsis button) the name of the main source code file.

Target File
Type in (or select with the Open File dialog after pressing the ellipsis button) the name of the target file (such as MyFile.EXE).

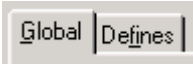
Project File
Type in (or select with the Open File dialog after pressing the ellipsis button) the name of the project file.

If you want to specify a working directory other than the one with which you started up the development environment, navigate the directory tree using the Open File dialog. When you've selected the directory you wish, type the project file name in the File Name box in the Open File dialog.

The Project File must be in the same directory as the main source file.

Target Type
Specify .EXE, .LIB or .DLL from the drop down list. The target file name will automatically add the correct extension.

Global Options / Compile Options Dialogs



These dialogs allow you to set compile options for the project, or its components. The options vary according to the item selected in the Project Tree dialog at the time you press the **Properties** button.

When you select a "folder level" item (such the project itself), the **Global Options** dialog appears. This allows you to set compile options for the project.

When you select a source code file, the **Compile Options** dialog appears. This provides the same options as the **Global Options** dialog; however, the selections apply only to the selected file. The **Compile Options** dialog also does not allow you to select the Build Mode, since that applies to the entire project.

The **Global Options** and **Compile Options** dialog contain the following options.

Global

Title — A short text description of the project. The Project System will list the description next to the Project name in the Project Tree list.

Target Type — Specify the type of executable file: choose **.EXE, .LIB**, or **.DLL** from the **Target Type** drop down list.

Run-Time Library — Specifies how the runtime library is called by the target file: choose ***Standalone***, ***Local***, or ***External*** from the **Run-Time Library** drop down list.

- ***Standalone*** — Uses the Clarion runtime library DLL (and database driver(s) .DLLs). It is called C60RUNx.DLL.
- ***Local*** — Links the runtime library and any database drivers into your executable using Smart Method Linking (only the necessary portions are linked in). This creates a "one-piece" executable.
- ***External*** — Specifies that another External DLL contains the runtime libraries and database drivers. The calls to this DLL must be exported.

Build Release System

To create an executable for release, check this box. To create an executable for use with the Debugger, uncheck this box, and set the Debug Mode option (located on the Debug tab control) to ***Full***.

Debug Information

Debug Mode Specifies the level of debug capability, choose ***Off***, ***Min***, or ***Full*** from the **Mode** drop down list.

Line Numbers Builds line numbers into the object file. This is not necessary for the Clarion debugger, but may be helpful when using other debuggers.

Runtime Checks

Stack Overflow Enables stack overflow warnings at runtime.

NIL-Pointer Allows compiler warnings when dereferencing null pointers.

Array Index Enables "array index larger than the array size" warnings at runtime.

Link Options

Stack Size To specify the stack size, type a value in Kilobytes in the **Stack Size** field.

Create Map File Creates a map file, which contains information about segment sizes and public functions. The map file may be used with third party debuggers.

Defines

Defines

To define a switch, or switches, for use with the COMPILE and OMIT compiler directives, type a list of valid Clarion labels separated by commas. Each label defines a separate switch.

For example, type 'Demo' in the **Defines** field. The Project System will create a switch called Demo and turn it "on." Now you can use the switch in conditional COMPILE and OMIT statements within your source code. For example:

```
COMPILE('END COMPILE',DEMO=ON)
 IF TODAY() > FirstRunDate + 30
  #ReturnCode = MESSAGE('Beta period expired')
  RETURN
 END
END COMPILE
```

Project Tree Dialog

The **Project Tree** dialog organizes all the components, and provides access to other dialogs that set additional options.

The Project file tracks all the components that make up the final executable file. It also sets the compiler options ranging from whether to include debug code or not, to setting a preferred optimization method.

If you use the Application Generator to create your source code, the only thing you will probably use the Project System for is to set debugging options.

Tree List

The Tree List itemizes the file level elements which comprise your project, including source code files, file drivers, other projects to compile, external libraries and resources, and other programs to execute as part of the make process.

Properties

Calls a dialog allowing you to specify compile options for the selected item, or the entire project. The particular dialog which appears depends on the currently selected item.

When you select a "folder level" item (such the project itself), the **Global Options** dialog appears. This allows you to set compile options for the project.

When you select a source code file, the Compile Options dialog appears. This provides the same options as the Global Options dialog; however, the selections apply only to the selected file.

Add File

Calls the Open File dialog, allowing you to add a file, such as a source code file, below the currently selected item.

By inserting a .PRJ file after the **Projects to Include** item, you can specify that the other project is built in the course of building the current project.

By inserting .ICO files after the **Application icon** item, you can link the icons into your executable so they do not have to be shipped separately.

Tip

.ICO files referenced within data structures are automatically linked into the executable.

For hand-coded applications, you can insert new .CLW files after the **External Source Files** item.

By inserting a.LIB file after the **Library, object, and resource files** item, you specify that the Project System should link it in to the project. By inserting a .DLL file, you specify that the application dynamically calls external functions from the file at run time.

You can also add external resources, such as .CUR, or .ICO files if they were not explicitly named as attributes in your source code. For example, if you specify variable naming a bitmap file in an IMAGE control (such as !MyBMP.BMP), you can link it in by adding it to the Project Tree below this item.

By inserting an executable file (*.exe, *.com, *.bat, or *.pif) after the **Programs to execute** item, you can run another application upon completion of the compile. This can be useful, for example, in network operations where you need to remap a drive after the compile, but before running the compiled application.

Remove File

Allows you to remove the currently selected item from the Project Tree. This does not physically remove the file from disk.

Edit

For hand-coded projects, if a source code file is selected, calls the Text Editor and loads it into a source code document window.

Make Statistics Dialog

This dialog displays the creation statistics of your latest compile.

For each object file created, it lists the object file name, source code file, size of the code and data segments, and the current date for each.

Create New Project

This dialog allows you to create a new project file. Mark the radio button for the type of project you wish to create.

Quick Start	Calls the Quick Start Wizard.
Application Generator	Calls the Application Properties dialog.
Hand Coded Project	Calls the New Project dialog.
Working Directory	Allows you to specify the directory where the new project will be created.

Report Formatter IDE Support

The following section offers a breakdown of the parts of the Clarion Report Formatter by alphabetic sequence.

Break Properties

This dialog lets you add or edit the properties of a group break.

General

Use	This defines the USE attribute for the BREAK structure. Type a field equate label to reference the BREAK structure in executable code.
Label	Type a valid Clarion label, naming the BREAK structure.
Variable	Type a variable name, to generate a break when the value changes as you sequentially process the file.

Group breaks provide a means of breaking the data into sections and optionally providing subtotals. Each group gathers a set of data file records, all sharing the same value in the BREAK field. Within a report, you may visually separate these rows, and add a subtotal or summary information, usually below the group. Group breaks are also called group bands by some popular end user database applications.

The group break may contain the same elements as the report: a group HEADER, group DETAIL, and group FOOTER. These structures all print inside the DETAIL print area, each following the other by any offset specified in their AT attributes.

Though they print on the page at the same time, the application composes the group HEADER before the group DETAIL. The group HEADER is a good place to identify the group.

The group FOOTER, is composed after the group DETAIL. You can place a string saying "Total:" followed by a string variable which contains the field to be summed, with the SUM attribute.

To create a group break:

1. Be sure the DETAIL band is visible; if not, press the restore button.
2. Choose **Bands** ▸ **Surrounding Break**.
3. When the cursor changes to a crosshair, CLICK in the DETAIL.
   This inserts the group BREAK.
4. In the **Break Properties** dialog, type the name of a variable or field, including the prefix, to break on.
5. Type a valid Clarion label to name the break.
6. Press the **OK** button.

When the report prints, it groups all records with the same value for the BREAK field, as well as the group HEADER and FOOTER.

Tip

If the break variable is a global or local variable, you must be sure that the executable code updates its value, so that it can generate a group BREAK.

Actions

Tables	Accesses the **Table Schematic Definition** dialog for this procedure.
Embeds	Accesses the **Embedded Source** dialog for this procedure.

Break Group Header Properties

This dialog lets you edit the properties of the group HEADER. To specify an element to compose at the start of each group, place it in the group HEADER.

Though they print on the page at the same time, the application composes the group HEADER before the group DETAIL. The group HEADER is a good place to identify the group, for example, with a label saying "Customer:" followed by a variable string for the customer name field.

General

Use

This defines the USE attribute for the structure. Type a field equate label to reference the HEADER in executable code.

Page Before

To print the HEADER structure on a new page, type a value in the **Page Before** box in the **Detail Properties** dialog. This sets the PAGEBEFORE attribute. The print engine generates a page break before printing at the top of the next page.

The page number automatically increments, unless you reset it. To reset the page number to a value you specify, type it in the **Page Before** field.

Page After

To print the HEADER, then force a new page, type a value in the **Page after** box. This sets the PAGEAFTER attribute. This prints the HEADER, then begins a new page.

The page number automatically increments, unless you reset it. To reset the page number to a value you specify, type it in the **Page after** field.

With Prior

To prevent 'orphan' elements in a printout, type a value in the **With Prior** field. This sets the WITHPRIOR attribute. An 'orphaned' print element is one which prints on a following page, separated from its related items.

The value specifies the number of preceding elements to print--a value of "1," for example, specifies that the previous element must print on the same page.

With Next To prevent 'widow' elements in a printout, type a value in the **Keep Next**. This sets the WITHNEXT attribute. A 'widowed' print element is one which prints, but then is separated from the succeeding elements by a page break.

The value specifies the number of succeeding elements to print--a value of '1,' for examples, specifies that the next element must print on the same page, else page overflow puts them on the next.

Freeze "Freezes" all the controls on the band so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color or Background fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your header declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Alone Specifies that the HEADER section always prints alone on a page.

Absolute Specifies that the HEADER section always prints at the same fixed position on the page. This adds the ABSOLUTE attribute to the DETAIL structure.

Position

Lets you set the location and size of the group header. The position is relative to the top left corner of the paper.

The **Position** dialog lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position visually, by dragging with the mouse in the **Report Formatter's Page Layout View**.

To set a location or size for your report section, specify fixed coordinates with this dialog. Choose from the following options for the **Top Left Corner**, the **Width**, and the **Height**.

Default Sets a value based on the Paper Size and Print Detail position (see also **Report Properties** dialog).

Full The header is the full width of the print detail area.

Fixed To set a specific position and size, mark the **Fixed** choices and enter a value in the entry box.

The measurement units for these boxes are specified on the **General** tab of the **Report Properties** dialog.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for all the controls in the report section. You may override the section font by setting a different font in the Properties dialog for any specific control. As you choose options, the dialog box displays a sample of the selected font.

Actions

Tables Accesses the **Table Schematic Definition** dialog for this procedure.

Embeds Accesses the **Embedded Source** dialog for this procedure.

Break Group Footer Properties

This dialog lets you edit the properties of the group FOOTER.

The group FOOTER is composed immediately after the group DETAIL, and provides the logical place for adding subtotals to your report.

General

Use

This defines the USE attribute for the structure. Type a field equate label to reference the FOOTER in executable code.

Page Before

To print the FOOTER structure on a new page, type a value in the **Page Before** box. This sets the PAGEBEFORE attribute. The print engine generates a page break before printing at the top of the next page.

The page number automatically increments, unless you reset it. To reset the page number to a value you specify, type it in the **Page Before** field.

Page After

To print the FOOTER, then force a new page, type a value in the **Page after** box. This sets the PAGEAFTER attribute. This prints the FOOTER, then begins a new page.

The page number automatically increments, unless you reset it. To reset the page number to a value you specify, type it in the **Page after** field.

With Prior

To prevent 'orphan' elements in a printout, type a value in the **With Prior** field. This sets the WITHPRIOR attribute. An 'orphaned' print element is one which prints on a following page, separated from its related items.

The value specifies the number of preceding elements to print--a value of "1," for example, specifies that the previous element must print on the same page.

With Next

To prevent 'widow' elements in a printout, type a value in the **Keep Next**. This sets the WITHNEXT attribute. A 'widowed' print element is one which prints, but then is separated from the succeeding elements by a page break.

The value specifies the number of succeeding elements to print--a value of '1,' for examples, specifies that the next element must print on the same page, else page overflow puts them on the next.

Freeze "Freezes" all the controls on the band so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color or Background fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your header declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Alone Specifies that the FOOTER section always prints alone on a page.

Absolute Specifies that the FOOTER section always prints at the same fixed position on the page. This adds the ABSOLUTE attribute to the FOOTER structure.

Position

Lets you set the location and size of the group footer. The position is relative to the top left corner of the paper.

The **Position** dialog lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position visually, by dragging with the mouse in the **Report Formatter's Page Layout View**.

To set a location or size for your report section, specify fixed coordinates with this dialog. Choose from the following options for the **Top Left Corner**, the **Width**, and the **Height**.

Default Sets a value based on the Paper Size and Print Detail position (see also **Report Properties** dialog).

Full The footer is the full width of the print detail area.

Fixed To set a specific position and size, mark the **Fixed** choices and enter a value in the entry box.

The measurement units for these boxes are specified on the **General** tab of the **Report Properties** dialog.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for all the controls in the report section. You may override the section font by setting a different font in the Properties dialog for any specific control. As you choose options, the dialog box displays a sample of the selected font.

Actions

Tables	Accesses the **Table Schematic Definition** dialog for this procedure.
Embeds	Accesses the **Embedded Source** dialog for this procedure.

Detail Band Properties

This dialog lets you edit the properties of the report detail.

General

Label Type a valid Clarion label, naming the DETAIL structure.

Use This defines the USE attribute for the structure. Type a field equate label to reference the Detail in executable code.

Page Before To print the DETAIL structure on a new page, type a value in the **Page Before** box in the **Detail Properties** dialog. This sets the PAGEBEFORE attribute. The report prints the full DETAIL starting at the top of the next page. The report FOOTER, however, prints on the first page.

The page number automatically increments, unless you reset it. To reset the page number to a value you specify, type it in the **Page Before** field.

Page After To print the DETAIL, then force a new page, type a value in the **Page after** box in the **Detail Properties** dialog. This sets the PAGEAFTER attribute. This prints the DETAIL, then prints the page FOOTER, then begins a new page.

Tip

To print a separate page for each record, place the variable strings and/or controls you wish in the DETAIL, and specify the PAGEAFTER attribute in the Detail Properties dialog.

The page number automatically increments, unless you reset it. To reset the page number to a value you specify, type it in the **Page after** field in the **Detail Properties** dialog.

With Prior To prevent 'orphan' elements in a printout, type a value in the **With Prior** field. This sets the WITHPRIOR attribute. An 'orphaned' print element is one which prints on a following page, separated from its related items.

The value specifies the number of preceding elements to print--a value of "1," for example, specifies that the previous element must print on the same page.

Tip

When placing subtotals or totals in a DETAIL, use the WITHPRIOR attribute to insure they print with at least one member of the column above it when a page break occurs.

With Next

To prevent 'widow' elements in a printout, type a value in the **Keep Next**. This sets the WITHNEXT attribute. A 'widowed' print element is one which prints, but then is separated from the succeeding elements by a page break.

The value specifies the number of succeeding elements to print--a value of '1,' for examples, specifies that the next element must print on the same page, else page overflow puts them on the next.

Freeze

"Freezes" all the controls on the band so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color or Background fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your header declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Alone

Specifies the print engine should print *only* the DETAIL section, without FORM, HEADER, and FOOTER sections. This setting is most useful for report title and grand totals pages. This adds the ALONE attribute to the DETAIL structure.

Absolute

Specifies that the DETAIL section always print at the same fixed position on the page. This adds the ABSOLUTE attribute to the DETAIL structure. Otherwise, the DETAIL prints at a position relative to the last section printed in the detail print area.

Together

Specifies that the band is to only print on a single page, rather than being automatically split if there is not sufficient room for it. This adds the TOGETHER attribute.

Position

Lets you set the location and size of the print detail, by filling in the AT attribute. The measurement units for these boxes are specified on the **General** tab of the **Report Properties** dialog.

To set a precise starting point for your detail relative to the last detail printed, specify **Top Left Corner** coordinates with this dialog. These settings may also be accomplished visually by dragging the report section and it's borders in **the Report Formatter's Page Layout View**.

To set the size of the detail, choose from the following options for the **Width** and **Height**.

Default	Sets a value based on the Paper Size.
Full	The detail is the full width of the detail print area.
Fixed	To set a specific size, mark the **Fixed** choices.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for all the controls in the report section. You may override the section font by setting a different font in the Properties dialog for any specific control. As you choose options, the dialog box displays a sample of the selected font.

Actions

Tables	Accesses the **Table Schematic Definition** dialog for this procedure.
Embeds	Accesses the **Embedded Source** dialog for this procedure.

Grid Size Dialog

This allows you to set the spacing between grid points. When you place a new control in the report, you can then optionally force it to the grid points with the **Option ▸ Snap to Grid** command.

Set the spacing by typing in values for the **Horizontal** and **Vertical** spacing. The measurement units depend on the default for the report, as set in the **Report Properties** dialog. The **Grid Size** dialog tells you the minimum value (1/20th inch, or 2 millimeters).

Page Header Properties

This dialog lets you edit the properties of the page HEADER. To specify an element to compose at the start of each report page, place it in the page HEADER.

Though they print on the page at the same time, the print engine composes the page HEADER before the page DETAIL, or FOOTER. The page HEADER is a good place for company logos, print dates, etc.

General

Use This defines the USE attribute for the structure. Type a field equate label to reference the page HEADER in executable code.

Freeze "Freezes" all the controls on the band so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color or Background fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your header declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Alone Has no effect on a Page Header.

Absolute Has no effect on a Page Header.

Together Specifies that the band is to only print on a single page, rather than being automatically split if there is not sufficient room for it. This adds the TOGETHER attribute.

Position

Lets you set the location and size of the page header. The position is relative to the top left corner of the paper.

The **Position** dialog lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position visually, by dragging with the mouse in the **Report Formatter's Page Layout View**.

To set a location or size for your report section, specify fixed coordinates with this dialog. Choose from the following options for the **Top Left Corner**, the **Width**, and the **Height**.

Default	Sets a value based on the Paper Size.
Full	The header is the full width of the paper.
Fixed	To set a specific position and size, mark the **Fixed** choices and enter a value in the entry box. The measurement units for these boxes are specified on the **General** tab of the **Report Properties** dialog.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for all the controls in the report section. You may override the section font by setting a different font in the Properties dialog for any specific control. As you choose options, the dialog box displays a sample of the selected font.

Actions

Tables	Accesses the **Table Schematic Definition** dialog for this procedure.
Embeds	Accesses the **Embedded Source** dialog for this procedure.

Page Footer Properties

This dialog lets you edit the properties of the page FOOTER. To specify an element to compose at the end of each report page, place it in the page FOOTER.

Though they print on the page at the same time, the print engine composes the page HEADER before the page DETAIL, or FOOTER. The page FOOTER is a good place for page numbers, page totals, etc.

General

Use

This defines the USE attribute for the structure. Type a field equate label to reference the page FOOTER in executable code.

Freeze

"Freezes" all the controls on the band so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color or Background fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your header declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Alone

Has no effect on a Page Footer.

Absolute

Has no effect on a Page Footer.

Together

Specifies that the band is to only print on a single page, rather than being automatically split if there is not sufficient room for it. This adds the TOGETHER attribute.

Position

Lets you set the location and size of the page footer. The position is relative to the top left corner of the paper.

The **Position** dialog lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position visually, by dragging with the mouse in the **Report Formatter's Page Layout View**.

To set a location or size for your report section, specify fixed coordinates with this dialog. Choose from the following options for the **Top Left Corner**, the **Width**, and the **Height**.

Default	Sets a value based on the Paper Size.
Full	The footer is the full width of the paper.
Fixed	To set a specific position and size, mark the **Fixed** choices and enter a value in the entry box. The measurement units for these boxes are specified on the **General** tab of the **Report Properties** dialog.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for all the controls in the report section. You may override the section font by setting a different font in the Properties dialog for any specific control. As you choose options, the dialog box displays a sample of the selected font.

Actions

Tables	Accesses the **Table Schematic Definition** dialog for this procedure.
Embeds	Accesses the **Embedded Source** dialog for this procedure.

Page Form Properties

To specify a static page element which prints on every page, place it in the FORM. This is a free-floating section which can overlap the other sections.

Use the FORM as a layer, to 'hold' graphic frames or preprinted *forms* into which the data from the other sections 'fit.' Another use for the FORM is to hold a 'watermark,' which prints underneath the report.

The FORM size defaults to the same size as the page, less the margins.

The print engine composes the FORM at the beginning of the print job; and does not update it thereafter. Therefore, the FORM is not suitable for holding variable data fields, or even a page number. You can, however, print fields from a control file, if you wish to print the same field contents on every page of the report.

Tip

For best results when using a drawing tool to create a 'watermark,' on, for example, for a 300 DPI printer, set the fill for the watermark element to 10% gray, or light gray. At higher printing resolutions, try 20% gray.

The FORM should guide the user to the data. You might use lines and boxes, for example, to divide the DETAIL into 'compartments,' grouping data and columns for the user. You may even create a 'greenbar paper' effect by alternating gray or light green color blocks.

General

Use	This defines the USE attribute for the structure. Type a field equate label to reference the FORM in executable code.
Freeze	"Freezes" all the controls on the band so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color or Background fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your header declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Position

Lets you set the location and size of the page form. The position is relative to the top left corner of the paper.

The **Position** dialog lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position visually, by dragging with the mouse in the **Report Formatter's Page Layout View**.

To set a location for the **Top Left Corner** of your form, specify **X** (horizontal) and **Y** (vertical) coordinates with this dialog. **X** = top margin and **Y** = left margin.

To set a size for your page form, choose from the following options for the **Width** and **Height**.

Default Sets a value based on the Paper Size.

Fixed To set a specific size, mark the **Fixed** choices and enter a value in the entry box.

The measurement units for these boxes are specified on the **General** tab of the **Report Properties** dialog.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for all the controls in the report section. You may override the section font by setting a different font in the Properties dialog for any specific control. As you choose options, the dialog box displays a sample of the selected font.

Actions

Tables Accesses the **Table Schematic Definition** dialog for this procedure.

Embeds Accesses the **Embedded Source** dialog for this procedure.

Preview Print Details Dialog

This allows you to choose a report section to preview.

Because you can nest many sections of various types within a single report, you have to select a section before actually previewing it. This way, the Report Formatter knows what parts you want to compose on the screen.

Select a section from the **Details** list, then press the **Add** button to move it to the **Selected Details** list. Press the **OK** button to preview.

Report Properties

This dialog lets you set up the basic report options, including its page orientation, measurement units, margins, and paper size.

General

Job Name Names the print job, as listed in the Windows Print Manager application.

Label Type a valid Clarion label to name the REPORT data structure.

Prefix Specifies the label prefix for the REPORT structure.

Units Specifies the default measurement for all controls placed in the report. Choose **Dialog Units**, **thousandths of Inches**, **Millimeters** or **Points**.

Note:

After changing the report Units, the Position tab is disabled. You must exit the Report Properties dialog to save the change of units. You can then reenter the Report Properties dialog to adjust the report position.

Freeze "Freezes" all the controls on the report so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color or Background fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your report declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Preview Specifies the name of a QUEUE which stores the filename(s) (*.WMF) for the metafile(s) generated for page preview. See the PREVIEW attribute. If you are using the Report Template, it is handled automatically if you check the Print Preview box and you should leave this entry blank.

Position

Lets you set the location and size of the report detail print area, by filling in the AT attribute. The measurement units for these boxes are specified on the **General** tab.

To set a precise starting point for your print detail area relative to the top left corner of the paper, specify **Top Left Corner** coordinates with this dialog. In effect, this establishes the left **margin** for your report. The top margin is usually determined by the Page Header position. These settings may also be accomplished visually by dragging report sections and borders in **the Report Formatter's Page Layout View**.

To set the size of the print detail area, choose from the following options for the **Width** and **Height**. When changing a report from portrait to landscape, or vice versa, you should also change the width and height on this tab.

Default	Sets a value based on the Paper Size.
Fixed	To set a specific size, mark the **Fixed** choices.

Paper Size

Paper Size	Choose from over 40 standard sizes, or choose ***Other*** to specify a custom size.
Width	Specifies a custom paper width in units specified on the General tab.
Height	Specifies a custom paper height in units specified on the General tab.
Landscape	Specifies landscape paper orientation. New reports default to portrait mode. Landscape means the report text is aligned parallel with the longest paper edges.

Font

To set the default font for all controls appearing in the report, press the **Font** button, then choose the font and style in the **Select Font** dialog. You may override the default by setting a different font in the Properties dialog for any specific control. The options you choose in the dialog become the parameters for the FONT attribute. As you choose options, the dialog box displays a sample of the formatting.

Actions

Tables	Accesses the **Table Schematic Definition** dialog for this procedure.
Embeds	Accesses the **Embedded Source** dialog for this procedure.

Using the Report Formatter - Align Toolbox

The **Report Formatter's Align** toolbox allows you to quickly, professionally, and precisely align the controls in your report.

Display or hide the **Align** toolbox by choosing **Options ▸ Alignbox.** Resize the **Align** toolbox by placing the cursor on the border of the box. When the cursor changes to a double-headed arrow, CLICK and DRAG.

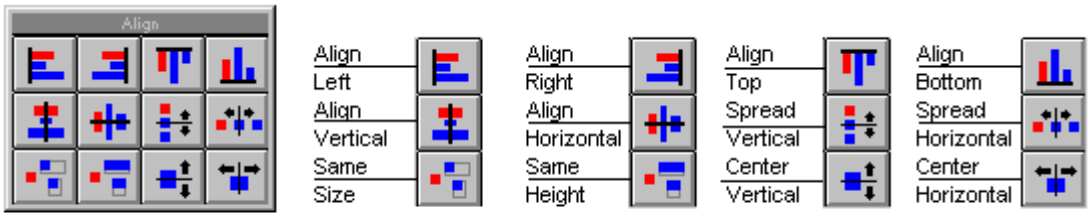
Select the controls to align (CTRL+CLICK allows you to select multiple controls, or you can "lasso" multiple controls with CTRL+DRAG), then click on the appropriate alignment tool. All the alignment actions are also available from the **Alignment** menu.

Tip

For most alignment functions, the first controls selected (blue handles) are aligned with the last control selected (red handles). That is, the last control selected is the anchor control. It doesn't move, the others do.

Tip

Position the cursor over any tool and wait for half a second. A tool tip appears telling you the type of alignment this tool will accomplish.



Align Left	Aligns the left borders of the selected controls with the left border of the last control selected (red handles).
Align Right	Aligns the right borders of the selected controls with the right border of the last control selected (red handles).
Align Top	Aligns the top borders of the selected controls with the top border of the last control selected (red handles).

Align Bottom	Aligns the bottom borders of the selected controls with the bottom border of the last control selected (red handles).
Align Vertically	Along a vertical axis, aligns the centers of the selected controls with the center of the last control selected (red handles).
Align Horizontally	Along a horizontal axis, aligns the centers of the selected controls with the center of the last control selected (red handles).
Spread Vertical	Equalizes the vertical spaces between the selected controls.
Spread Horizontal	Equalizes the horizontal spaces between the selected controls.
Same Size	Makes all selected controls the same height and width as the last control selected (red handles).
Same Height	Makes all selected controls the same height as the last control selected (red handles).
Center Vertically	As a group (relative positions of selected controls don't change), centers the selected controls horizontally within the band.
Center Horizontally	As a group (relative positions of selected controls don't change), centers the selected controls vertically within the band.

Using the Report Formatter - Command Toolbox

The **Report Formatter** contains a dockable **Command** toolbox. The toolbox lets you quickly execute a variety of **Report Formatter** functions at the touch of a button.

All the commands in the **Command** toolbox are also available from the menu (**Exit!, Edit, View, Preview!**).

Display or hide the **Command** toolbox by choosing **View ▸ Show Commandbox.** Resize the **Command** toolbox by placing the cursor on the border of the box. When the cursor changes to a double headed arrow, CLICK and DRAG. Dock the toolbox by dragging the handle (double vertical lines) to any edge of the Window Formatter frame (dragging the title bar repositions the toolbox *without* docking).

Tip: Position the cursor over any tool and wait for half a second. A tool tip shows you the type of control this tool creates.

Save Changes	Exit the Window Formatter and save changes.
Lose Changes	Exit the Window Formatter and abandon changes.
Properties	Edit the properties of the selected (red handles) control or window.
Show Toolbox	Hide or display the Controls Toolbox.
Show Alignbox	Hide or display the Align Toolbox.
Show Property Box	Hide or display the Property Toolbox.
Show Column Box	Hide or display the Fields Toolbox.
Grid	Toggle the alignment grid on or off.
Preview	Preview the window.

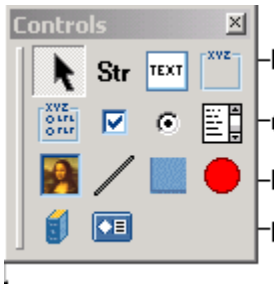
Using the Report Formatter - Controls Toolbox

The **Report Formatter** contains a floating **Controls** toolbox, similar to the **Window Formatter**. Simply choose a control from the toolbox (CLICK on it), then CLICK in the report band to place the control in the report.

Display or hide the **Controls** toolbox by choosing **Options ▸ Toolbox.** Resize the **Controls** toolbox by placing the cursor on the border of the box. When the cursor changes to a double headed arrow, CLICK and DRAG. All the controls in the toolbox are also available from the **Controls** menu.

Tip

Position the cursor over any tool and wait for half a second. A tool tip appears telling you the type of control that will be created by this tool.



No Tool	String	Text Box	Group
Option	Check box	Radio Button	List Box
Image	Line	Box	Ellipse
Dictionary Field	VBX	Control Template	

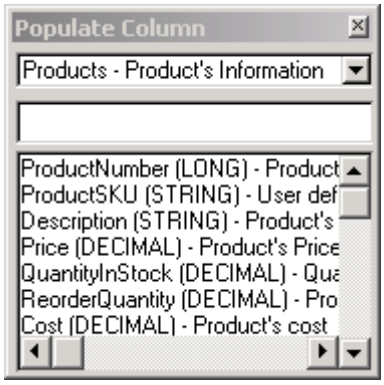
String	Allows you to place STRING control on the report under construction.
Text Box	Allows you to place TEXT control on the report under construction.
Group Box	Allows you to place GROUP control (group box) on the report under construction.
Option Box	Allows you to place OPTION control (OPTION structure, which appears as a group box with radio buttons) on the report under construction.
Check Box	Allows you to place CHECKBOX control on the report under construction.
Radio Button	Allows you to place RADIO control on the report under construction.
List Box	Allows you to place LIST control (list box, or drop down list box) on the report under construction.

Image	Allows you to place IMAGE control (graphic image) on the report under construction.
Line	Allows you to place LINE control on the report under construction.
Rectangle	Allows you to place a BOX control on the report under construction.
Ellipse	Allows you to place ELLIPSE control on the report under construction.
Dictionary Field	Allows you to select a field defined in the Data Dictionary, and place the control specified in the data dictionary, plus an associated PROMPT control, on the report under construction.
Custom Control	Allows you to place a VBX control (Visual Basic custom control) on the report under construction.
Control Template	Allows you to place Control Template on the report under construction.

Using the Report Formatter - Populate Columns

The **Report Formatter** contains a floating **Populate Columns** toolbox. This toolbox allows you to quickly "populate" a window with entry controls and prompts for columns in your data dictionary table.

Display or hide the **Populate Columns** toolbox by choosing **Options ▸ Show Column Toolbox.** Resize the **Populate Columns** toolbox by placing the cursor on the border of the box. When the cursor changes to a double-headed arrow, CLICK and DRAG.



1. Choose a **table** from the drop down list.
2. Use the locator to optionally search for a column. CLICK on the **column** you want on your report.
3. CLICK in the report band to place the control.

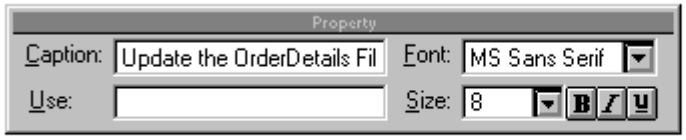
The cursor becomes a crosshair. The top left corner of the control is placed at the intersection of the cursor crosshair.

The **type** of control (text box, check box, radio button, etc.) is determined by the settings for this particular column in the Data Dictionary **Column Properties** dialog.

Using the Report Formatter - Property Toolbox

The **Report Formatter's Property** toolbox allows you to quickly specify the appearance and content of the text on each in the report. Control the font, size, style, and content of all your text, using standard word processor buttons and drop down lists. **Select Font** dialog.

Display or hide the **Property** toolbox by choosing **Options ▸ Propertybox.** Resize the **Property** toolbox by placing the cursor on the border of the box. When the cursor changes to a double headed arrow, CLICK and DRAG.



Text Formatting: choose content, font, size, and style for the selected control or window.

Using the Report Formatter - Sample Reports

The **Report Formatter** is a visual design tool. In **Band View**, you always see a sample of the report band you're working on, as you work on it. For example, place a list box in the detail band and drag its handles to the size you want.

Switch to **Page Layout View** to resize and reposition the report bands. Drag the band handles or drag the entire band. All bands appear together on a representation of the page.

In addition, you can quickly generate filler data and see a sample report by choosing **Preview!** from the action bar, all without actually compiling or running the report.

Text Editor

The following section describes the features of the IDE's Text Editor.

Text Editor Menu Commands

The Text Editor is a full function programmer's editor featuring Multiple Document Interface support, auto-indent, search-and-replace, and color-coded syntax highlighting.

Related Topics: Redirection File

Use the floating Fields toolbox to insert fully qualified variable and field names at the insertion point.

The following lists all menu commands available from within the Text Editor.

You can also get help for a keyword within a document window by placing the insertion point on the keyword, and pressing the F1 key. This lets you quickly look up help for a Clarion language statement, function or attribute.

Note that some of the commands, most notably on the Project and Setup menus, do not specifically reference Text Editor functions. Because the Project System and the Registries are always active, their menu commands are always available.

File Menu

New	Opens the **New** dialog, which lets you create a new source code or other type of file.
Open	Calls the **Open File** dialog, allowing you to open a source code document.
Pick	Calls the **Pick** dialog, listing the most recently used files in a list box.
Close	Closes the active source code document.

Quick Start New App Calls the Clarion Quick Start dialog window. You will first access the New dialog window, to designate the name, and where you want to create it.

Save Saves the active source code document.

Save As Saves the active source code document under a new name which you specify.

Save All Saves all the currently open source code documents.

Print Prints the active source code document.

Print Setup Calls the **Printer Setup** dialog, allowing you to configure your printer.

Import File Calls the **Open File** dialog, allowing you to insert the contents of a file into the active source code document, at the insertion point.

Export Block Saves the selected text in a new source code document under a new name which you specify.

Browse Database Browses and edits a database file defined in the current dictionary. Select the file from the **Pick** file dialog, or the **Open File** dialog after specifying a database driver.

Convert Application

The **Convert Application** command starts the Clarion Application Conversion Wizard to convert applications developed in one Clarion environment to another Clarion environment (newer Clarion version, different templates, etc.). See *Application Converter* for more information.

Note: If your templates are unchanged, you can open application files with newer versions of Clarion and the file conversion is automatic. The Application Conversion Wizard carries out more complex conversions.

Exit Quits the program.

Edit Menu

Undo Reverses the most recent editing action.

Cut Deletes the selected text from the document and places it in the clipboard.

Copy Places a copy of the selected text from the document into the clipboard.

Paste Pastes text from the clipboard into the active document, at the insertion point.

Select All Selects all text in the active document.

Goto Line Calls the **GoTo Line** dialog, in which you can enter a line number. After pressing the **OK** button, the cursor moves to the beginning of the line you specify.

Goto Next Error Moves the insertion point to the next compiler error. The error appears at the bottom of the window. This command is disabled except following a compile which generated errors.

Goto Previous Error Moves the insertion point to the previous compiler error. The error appears at the bottom of the window. This command is disabled except following a compile which generated errors.

Set/Clear Tabstop Places a custom tab stop at the insertion point.

Duplicate Line Places a copy of the current line at the line immediately following it. You do not need to select the entire line. The insertion point merely needs to be anywhere within the line you wish to copy.

Toggle Case Changes the case of the next character following the insertion point.

Delete Line Deletes the entire line at which the insertion point is located.

Delete Word Deletes a current word at which the insertion point is located.

Format Structure Calls the **Window Formatter** or **Report Formatter** allowing you to edit or create a structure.

Search Menu

Find Calls the **Find** dialog. It lets you type in a word, then find the next occurrence forwards or backwards from the current position of the insertion point. The keyboard accelerator is ALT+F3.

Type the word or phrase to search for in the **Find What** box. Optionally indicate whether you wish the search to **Match Whole Word Only**, and/or **Match Case**. Choose a forward search (Down) or backwards search (Up), then press the **Find Next** button.

The **Find** dialog is modeless. This means that the dialog will remain on screen so that you may easily search again. This makes it easy to repeat a search several times quickly, using the **Find Next** button.

Replace

Calls the **Replace** dialog, which lets you change specific text to something else. You may make the changes one at a time, throughout a selected text block, or throughout the entire document.

Type the original word or phrase to replace in the **Find What** box. Type the replacement text in the **Replace With** box. Optionally indicate whether you wish the search to **Match Whole Word Only**, and/**or Match Case**.

Press the **Find Next** button to find the next occurrence of the word or phrase. Press the **Replace** button to replace it, once found. Press the **Replace All** button to replace all instances within the document.

The **Replace** dialog is modeless. This means that the dialog will remain on screen so that you may easily "replace" again. This makes it easy to repeat an operation several times quickly, using the **Find Next** button.

Find Next

Searches for the text most recently searched for, moving toward the top of the document.

Find Previous

Searches for the text most recently searched for, moving toward the beginning of the document.

Find Marked Text

Finds the next occurrence of the selected text. This is equivalent to executing the **Find** command, typing the selected text in the **Find What** box, and specifying a forward search.

Project Menu

Set

Calls the **Open File** dialog, allowing you to change the active .APP or .PRJ.

New

Calls the **New Project File** dialog, allowing you to create a new project.

Load

Opens the **Project Tree** dialog for hand coded projects, or **Application Tree** dialog for generated projects.

Edit

Opens the **Project Tree** dialog, allowing you to add or edit component files in the current project.

Make

Compiles and links the active application or project, which is named on the caption bar.

Make & Run

Compiles, links and runs (if a target executable) the active application or project, which is named on the caption bar.

Run Executes the active application or project, which is named on the caption bar.

Debug Loads the Debugger and prepares the active application or project, listed on the caption bar, for debugging.

Make Statistics Calls the **Make Statistics** dialog. Lets you view a statistical profile of the most recent make.

Auto Make Before Run Toggles the Project System setting which forces a recompile each time you choose the Run command.

File Save Before Run Toggles the Project System setting which saves the source code file each time you choose the Run command.

Minimize on Run Toggles the Project System setting which minimizes CW before displaying the application each time you choose the Run command.

Wait for Termination on Run Toggles the Project System setting which suspends CW until after you terminate the application upon executing it with the Run command.

Setup Menu

Editor Options Calls the **Editor Options** dialog, which lets you customize the appearance and behavior of the Text Editor.

Dictionary Options Calls the **Dictionary Options** dialog, which lets you specify default settings for the Dictionary Editor.

Application Options Calls the **Application Options** dialog, which lets you specify default settings for the Application Generator.

Template Registry Calls the **Template Registry** dialog, which lets you register and manage templates.

Database Driver Registry Calls the **Database Driver Registry**, which lets you register database drivers.

Edit Redirection File Loads the Redirection File in a document window, ready for editing.

VCS Options Calls the **Configure VCS Interface** dialog, which lets you configure a selected Version Control System for your dictionaries and applications.

Window Menu

Tile Vertically	Arranges open document windows side by side in a vertical orientation.
Tile Horizontally	Arranges open document windows side by side in a horizontal orientation.
Cascade	Arranges open document windows in overlapped fashion so that all caption bars are visible
Arrange Icons	Arranges iconized windows along the bottom of the Clarion Application frame.
(Window List)	Lists all open document windows by their caption bar text according to the order they were opened. Choosing a window from the list brings the window to the top.

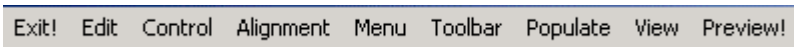
Help Menu

Contents	Opens the Windows Help application and displays a list of main topics.
Search for Help On	Opens the **Search** dialog in the Windows Help application, allowing you to search for help topics containing a specific keyword.
How to Use Help	Opens the Windows Help application and displays instructions for using the Help system.
About Clarion	Displays the program name, version, registration, and copyright information.

Window Formatter IDE Support

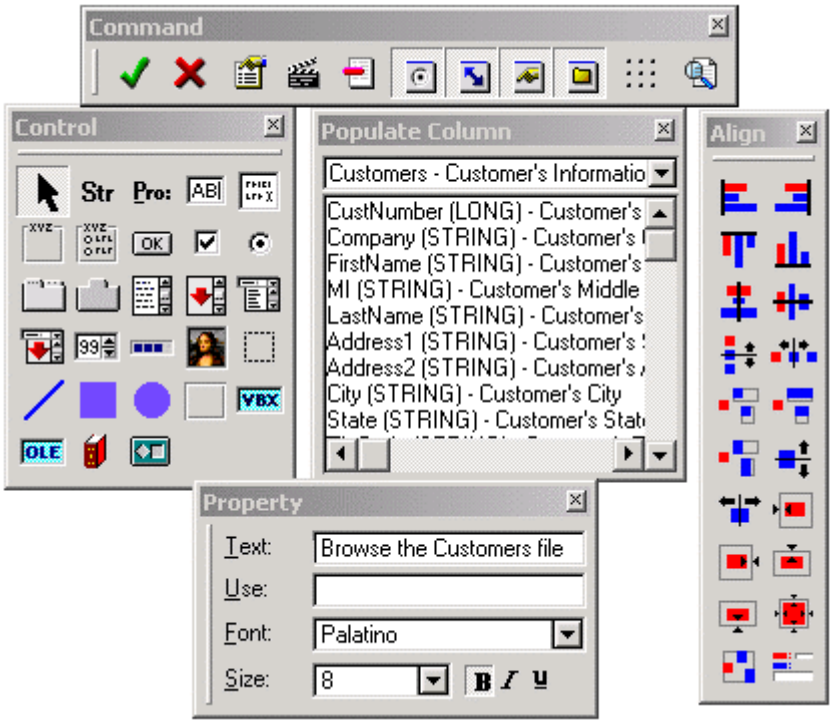
The following topics focus on special dialogs that are part of the Window Formatter as integrated within the Application Generator and the Text Editor.

Window Formatter Menu Commands



The **Window Formatter** helps you visually design Window elements--windows and controls--on screen. The **Window Formatter** automatically generates and places the language structures and source code that describe these elements in your .APP file or source code document.

Click on a toolbar shown below for more information.



Popup (right-click) Menu

The popup menu provides quick and easy access to a subset of the **Edit** menu commands.

Properties	Opens the **Properties** dialog for the selected window or control.
Actions	Opens the **Properties** dialog to the **Actions** tab for the selected control. Specify a variety of code options depending on the type of control and the template associated with the control.
Embeds	Opens the **Embedded Source** dialog for the selected window or control. Lets you manage embedded source at embed points associated with the window or control.
Font	Opens the **Select Font** dialog for the selected window or control. Choose typeface, size, style, color, etc. from standard drop down lists.
Key	Opens the **Input Key** dialog for the selected control. Establish a hot key, or key combination, that gives immediate focus to the control, or for buttons, initiates the button's action.
Alert	Opens the **Alert Keys** dialog for the selected control. Add or delete one or more keys, or key combinations, that will generate an EVENT:Alert when the control has focus.
Position	Opens the **Properties** dialog to the **Position** tab for the selected window or control. Specify default positioning, size, and/or exact x and y coordinates.
List Box Format	Opens the **List Box Formatter** for the selected list box control. Add, delete, resize, and reorder the fields in the list box. Format the fields or groups of fields.
Duplicate	Places a copy of the selected control or controls in the window under construction. Only the control is duplicated; any associated template code is not duplicated.
Delete	Deletes the selected control.
Custom	Opens the Property Sheet for an OCX associated with an OLE control.
Open	Opens the OLE Server associated with an OLE control.

Synchronize Applies the control attributes specified in the Data Dictionary to the selected control, or if the window is selected, to all the controls in the window. The attributes are applied as specified in the **Synchronization** tab of the **Application Options** dialog.

Exit!

Exits the **Window Formatter**. You are prompted to save or discard any changes.

Edit Menu

Undo Reverses the most recent editing action.

Redo Reverses the most recent undo action.

Properties Opens the **Properties** dialog for the selected window or control.

Actions Opens the **Properties** dialog to the **Actions** tab for the selected control. Specify a variety of code options depending on the type of control and the template associated with the control.

Embeds Opens the **Embedded Source** dialog for the selected window or control. Lets you manage embedded source at embed points associated with the window or control.

Font Opens the **Select Font** dialog for the selected window or control. Choose typeface, size, style, color, etc. from standard drop down lists.

Key Opens the **Input Key** dialog for the selected control. Establish a hot key, or key combination, that gives immediate focus to the control, or for buttons, initiates the button's action.

Alert Opens the **Alert Keys** dialog for the selected control. Add or delete one or more keys, or key combinations, that will generate an EVENT:Alert when the control has focus.

Position Opens the **Properties** dialog to the **Position** tab for the selected window or control. Specify default positioning, size, and/or exact x and y coordinates.

List Box Format Opens the **List Box Formatter** for the selected list box control. Add, delete, resize, and reorder the fields in the list box. Format the fields or groups of fields.

Duplicate	Places a copy of the selected control or controls in the window under construction. Only the control is duplicated, associated template code is not duplicated.
Delete	Deletes the selected control.
Custom	Opens the Property Sheet for an OCX associated with an OLE control.
Open	Opens the OLE Server associated with an OLE control.
Synchronize	Applies the control attributes specified in the Data Dictionary to the selected control, or if the window is selected, to all the controls in the window. The attributes are applied as specified in the **Synchronization** tab of the **Application Options** dialog.
Set Tab Order	Opens the **Ordering Type** dialog, which lets you visually specify the tab-stop order of the controls in the window. Manual - Select this radio button, then press the OK button to specify the tab-stop order by CLICK on the controls. A small box with a number inside appears on each control, indicating the current order. CLICK on the controls to change the order to the order you wish. Automatic - Select this radio button, then press the **OK** button to specify that the **Window Formatter** should set the tab-stop order based on the position of the controls. Choose **Horizontally** or **Vertically** from the options below. Reselect the **Set Tab Order** menu command to toggle back to normal editing mode.
Property Editor	Opens the Property Editor dialog for the window. This utility allows you to set properties on controls in the window and set the tab-stop order of the controls in the window by reordering a list of controls.
Synchronize Window	Applies the control attributes specified in the Data Dictionary to all the controls in the window. The attributes are applied as specified in the **Synchronization** tab of the **Application Options** dialog.
Control Templates	Opens the **Edit Control Templates** dialog, which lets you access the **Prompts** dialogs of any control templates in the window. This is equivalent to RIGHT-CLICKING a control template, then choosing **Actions** from the popup menu.

Control Menu

Push Button	Lets you place a BUTTON control on the window under construction. See also: Button Control Properties.
Radio Button	Lets you place a RADIO control on the window under construction. See also: Radio Button Control Properties.
Check Box	Lets you place a CHECK control on the window under construction. See also: Check Box Control Properties.
Entry Field	Lets you place an ENTRY control on the window under construction. See also: Entry Box Control Properties.
Text Field	Lets you place a TEXT control on the window under construction. See also: Text Box Control Properties.
Spin Box	Lets you place a SPIN control on the window under construction. See also: Spin Box Control Properties.
String	Lets you place a STRING control on the window under construction. See also: String Control Properties.
Prompt	Lets you place a PROMPT control on the window under construction. See also: Prompt Control Properties.
Group Box	Lets you place a GROUP control (group box) on the window under construction. See also: Group Control Properties
Option Box	Lets you place an OPTION control (OPTION structure, which appears as a group box with radio buttons) on the window under construction. See also: Option Control Properties.
List Box	Lets you place a LIST control on the window under construction. See also: List Control Properties.
Drop List Box	Lets you place a LIST control (with the DROP attribute) on the window under construction. See also: List Control Properties.
Combo Box	Lets you place a COMBO control on the window under construction. See also: Combo Box Control Properties.
Drop Combo Box	Lets you place a COMBO control (with the DROP attribute) on the window under construction. See also: Combo Box Control Properties.

Ellipse	Lets you place an ELLIPSE control on the window under construction. See also: Ellipse Control Properties.
Line	Lets you place a LINE control on the window under construction. See also: Line Control Properties.
Rectangle	Lets you place a BOX control on the window under construction. See also: Box Control Properties.
Image	Lets you place an IMAGE control (graphic image) on the window under construction. See also: Image Control Properties.
Region	Lets you place a REGION control on the window under construction. See also: Region Control Properties.
Panel Control	Lets you place a PANEL control on the window under construction. See also: Panel Control Properties.
Property Sheet	Lets you place a SHEET control on the window under construction. See also: Sheet Control Properties.
Tab Control	Lets you place a TAB control on the window under construction. See also: Tab Control Properties.
Progress Bar	Lets you place a PROGRESS control on the window under construction. See also: Progress Control Properties.
OLE/OCX Control	Lets you place an OLE control on the window under construction. See also: OLE Control Properties Dialog.

Alignment Menu

The Alignment menu provides commands for spacing and sizing the controls within the window. You may place two or more controls so that their 'edges' match up with each other. You may also spread the controls out, or make all of them the same size.

To do so, first select two or more controls. Select the first by clicking on it. Select the second and subsequent controls by pressing the CTRL key, then clicking on the second control while the shift key remains pressed. Lasso multiple controls by CTRL+CLICK+DRAGGING to form a box around the controls.

Set Left Margin Repositions the selected controls so the left edge of each control is a specified distance from the nearest bounding control or window. The margin distance toggles between two standard left margins. SHIFT+CLICK resizes the control instead of moving it. See *Configuring the Window Formatter--Margin Defaults*.

Set Right Margin Repositions the selected controls so the right edge of each control is a specified distance from the nearest bounding control or window. The margin distance toggles between two standard margins. SHIFT+CLICK resizes the control instead of moving it. See *Configuring the Window Formatter--Margin Defaults*.

Set Top Margin Repositions the selected controls so the top edge of each control is a specified distance from the nearest bounding control or window. The margin distance toggles between two standard top margins. SHIFT+CLICK resizes the control instead of moving it. See *Configuring the Window Formatter--Margin Defaults*.

Set Bottom Margin Repositions the selected controls so the bottom edge of each control is a specified distance from the nearest bounding control or window. The margin distance toggles between two standard margins. SHIFT+CLICK resizes the control instead of moving it. See *Configuring the Window Formatter--Margin Defaults*.

Set All Margins Repositions the selected controls so all edges of each control are a specified distance from the nearest bounding control or window. The margin distance toggles between two standard margins. SHIFT+CLICK resizes the control instead of moving it. See *Configuring the Window Formatter--Margin Defaults*.

Tip

Use the Set All Margins tool to position and size SHEET, GROUP, and LIST controls within their respective containers.

Shrink Wrap	If the primary (red) selection is a container control that surrounds all secondary (blue) selections, this repositions and resizes the primary selection to the minimum dimensions necessary to surround the secondary selections. Otherwise, this creates a new UBOXED (hidden) GROUP control minimally surrounding the selected controls. This lets you position the contained controls with the Margin Alignment tools.
Align Prompts	For selected prompt/field pairs, this aligns the prompt with its (visually) associated (ENTRY, SPIN, COMBO, TEXT, etc.) field.
Align Left	Aligns the left borders of the selected controls with the left border of the last control selected (red handles).
Align Right	Aligns the right borders of the selected controls with the right border of the last control selected (red handles).
Align Top	Aligns the top borders of the selected controls with the top border of the last control selected (red handles).
Align Bottom	Aligns the bottom borders of the selected controls with the bottom border of the last control selected (red handles).
Align Horizontally	Along a horizontal axis, aligns the centers of the selected controls with the center of the last control selected (red handles).
Align Vertically	Along a vertical axis, aligns the centers of the selected controls with the center of the last control selected (red handles).
Spread Horizontally	Equalizes the horizontal spaces between the selected controls.
Spread Vertically	Equalizes the vertical spaces between the selected controls.
Make Same Size	Makes all selected controls the same height and width as the last control selected (red handles).
Make Same Height	Makes all selected controls the same height as the last control selected (red handles).
Make Same Width	Makes all selected controls the same width as the last control selected (red handles).
Center Horizontally	As a group (relative positions of selected controls don't change), centers the selected controls vertically within the window.

Center Vertically As a group (relative positions of selected controls don't change), centers the selected controls horizontally within the window.

Set Left Margin Repositions the selected controls so the left edge of each control is a specified distance from the nearest bounding control or window. The margin distance toggles between two standard left margins. SHIFT+CLICK resizes the control instead of moving it. See *Configuring the Window Formatter--Margin Defaults*.

Set Right Margin Repositions the selected controls so the right edge of each control is a specified distance from the nearest bounding control or window. The margin distance toggles between two standard margins. SHIFT+CLICK resizes the control instead of moving it. See *Configuring the Window Formatter--Margin Defaults*.

Set Top Margin Repositions the selected controls so the top edge of each control is a specified distance from the nearest bounding control or window. The margin distance toggles between two standard top margins. SHIFT+CLICK resizes the control instead of moving it. See *Configuring the Window Formatter--Margin Defaults*.

Set Bottom Margin Repositions the selected controls so the bottom edge of each control is a specified distance from the nearest bounding control or window. The margin distance toggles between two standard margins. SHIFT+CLICK resizes the control instead of moving it. See *Configuring the Window Formatter--Margin Defaults*.

Set All Margins Repositions the selected controls so all edges of each control are a specified distance from the nearest bounding control or window. The margin distance toggles between two standard margins. SHIFT+CLICK resizes the control instead of moving it. See *Configuring the Window Formatter--Margin Defaults*.

Tip: Use the Set All Margins tool to position and size SHEET, GROUP, and LIST controls within their respective containers.

Shrink Wrap If the primary (red) selection is a container control that surrounds all secondary (blue) selections, this repositions and resizes the primary selection to the minimum dimensions necessary to surround the secondary selections. Otherwise, this creates a new UBOXED (hidden) GROUP control minimally surrounding the selected controls. This lets you position the contained controls with the Margin Alignment tools.

Align Prompts For selected prompt/field pairs, this aligns the prompt with its (visually) associated (ENTRY, SPIN, COMBO, TEXT, etc.) field.

Snap All To Grid

Snaps all controls to the nearest grid coordinate.

Snap To Grid

Snaps the selected controls to the nearest grid coordinate.

Menu Menu

New Menu Calls the Menu Editor, allowing you to create a menu for the window under construction.

Edit Menu Calls the Menu Editor, allowing you to edit an existing menu for the window under construction.

Delete Menu Lets you delete an existing menu for the window under construction.

Toolbar Menu

New Toolbar Adds a tool bar to the window under construction.

Delete Toolbar Deletes the existing tool bar for the window under construction.

Populate Menu

Field Places an entry control for a data dictionary field or memory variable, and an associated prompt. When you CHOOSE **Populate ▸ Field,** the **Table Schematic Definition** dialog appears. Select a field or variable, then CLICK in the window.

The CLICK places the prompt for the control as well as the control. If you pre-formatted the field, on the **Window** tab of the **Field Properties** dialog (for example, specifying a spin control), the control you specified appears, rather than an entry box.

Multiple Fields

Places an entry control for a data dictionary field or memory variable, and an associated prompt. When you choose **Populate ▸ Field,** the **Table Schematic Definition** dialog appears. Select a field or variable, then CLICK in the window.

The CLICK places the prompt for the control as well as the control. If you pre-formatted the field, on the **Window** tab of the **Field Properties** dialog (for example, specifying a spin control), the control you specified appears, rather than an entry box.

After placing the first field, the **Table Schematic Definition** dialog appears again, ready for you to select another field. When all fields are placed, press the **Cancel** button in this dialog to return to normal editing.

Control Template

Lets you add a control template to the window under construction. Select one from the **Select Control Template** dialog.

A control template adds a control or controls to the window, plus the code to maintain them. For example, the Browse Box control template places a list box in the window, lets you choose the fields for the list, and adds all the executable code for managing the list box.

Once the control template is placed, you can specify its properties and actions by RIGHT-CLICKING and selecting **Properties** or **Actions** from the popup menu, or choosing **Properties** or **Actions** from the **Window Formatter** menu.

View Menu

Options

Opens the Window Formatter Options window.

Show Grid

Toggles the Snap to Grid function on or off. Snap to Grid forces controls on the window to move in fixed increments set in the **Grid Settings** dialog.

Show Commandbox

Toggles display of the Command toolbox, which lets you access a variety of Window Formatter functions at the touch of a button.

Show Toolbox

Toggles display of the Controls toolbox, which lets you choose a control type and place it in a window. The tool icons available match those on the **Controls** menu and the **Populate** menus, described above.

Show Alignbox

Toggles display of the Align toolbox, which provides tool buttons for executing the align commands which appear on the **Alignment** menu, described above.

Show Propertybox Toggles display of the Property toolbox, which provides tools for setting some common control properties such as caption text, field equate label/use variable, and font.

Show Fieldbox Toggles display of the Fields toolbox, which places an entry control and prompt for a data dictionary field, when you DOUBLE-CLICK the field in the list box. The list contains all data dictionary fields defined in the **Table Schematic Definition** dialog for this procedure.

VBX Control Registry Opens the **VBX Custom Control Registry** dialog, which lets you manage the VBX controls available for use in the **Window Formatter**.

Preview!

Lets you display an active window identical to the one the end user sees. This lets you see how the window behaves, and how, for example, it looks with the 3D option set.

To exit **Preview** mode, press ESC, or press any DEFAULT button control in the previewed window.

APPLICATION (Frame) Properties Dialog

This dialog lets you specify the appearance and functionality of your application frame window.

General

Text

To specify caption bar text, type a string constant in the **Text** field. The caption bar holds the name of the window.

Label

To specify the label for the application structure, type it in the **Label** field. This names the specific APPLICATION in the source code. The label may contain upper or lower case letters, numerals, the underscore character or a colon. Space characters are forbidden. The first character must be a letter or the underscore character. Clarion reserved words may not serve as labels.

Frame Type

To choose the frame for your window, pick a selection from the **Frame Type** drop-down list. The frame defines the borders of the window. The normal frame type for an application frame is the resizable type. Choose from:

Single - a single pixel frame which the user cannot resize.

Double - a thick frame, which the user cannot resize. This adds the DOUBLE attribute to the window.

Resizeable - a thick frame, which the user can resize. This adds the RESIZE attribute to the window.

Layout

Indicates the orientation of window controls and field sequence.

Left to Right maintains the original layout specified in the Window Formatter.

Default field navigation moves from left to right.

Right to Left essentially "flips" the window controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child window that have the **default** setting active.

Initial Size

Sets the initial state of your window. Choose from:

Normal - displays the window at the default size which either you specifically set, or Windows sets if you don't.

Maximized - the window fills the desktop, if an application window, or the window frame, if an MDI child window. This adds the MAXIMIZE attribute to the window.

Iconized - the window appears in iconized state--as a 32 by 32 pixel window at the bottom of the desktop. This adds the ICONIZE attribute to the window.

Freeze

"Freezes" all the controls on the window so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color, Background, Selected text, or Selected fill fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your window declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Icon

To associate an icon with the window, specify an icon in this field. You may type in a file name or an EQUATE. You may also press the ellipsis button (...), then select an icon file name using the standard Open File dialog. The file name or equate you specify becomes the parameter for the ICON attribute.

You should always specify an icon for an application window. Specifying an icon name automatically places a minimize button on the caption bar of your application or MDI child window.

Timer

To specify the window receive Timer Event messages from Windows, fill in the **Timer** field. Specify the timer interval in hundredths of seconds. The file name or equate you specify becomes the parameter for the TIMER attribute.

For example, if you specify 100 in the field, the window will automatically receive an EVENT:Timer once every second (100/100's seconds). This might be appropriate for adding a clock to a status bar.

Immediate

To generate a message event each time the end user moves or resizes the window, check the **Immediate** box. This adds the IMM attribute to the window. You are responsible for the code that executes upon notification of the event.

Status Bar

To provide a message bar at the bottom of your window, mark the **Status Bar** check box. This adds the STATUS attribute to the window.

Tip

A status bar in an application window is an excellent way to increase user feedback in your application. Clarion makes it simple to post messages on the status bar advising the user of what your application is doing as it does it. Increasing user feedback makes the user feel more in control. This allows the user to feel more confident and be more efficient when using your application.

System Menu

To add a system menu to your window, mark the **System Menu** check box. Most windows should have a system menu. For users on a system without a mouse, the system menu provides the only means of minimizing, maximizing or re-sizing the window. This adds the SYSTEM attribute to the window.

Tip

Even if you plan that the window should NOT have a system menu when the application is complete, it's good practice to place a system menu on your application while it's under development. By DOUBLE-CLICKING the system menu, or choosing Close, you can close your application should your normal exit procedure fail.

Auto Display

To add the AUTO attribute to your window, mark the **Auto Display** check box. This automatically updates the contents of all controls on screen through each pass of the ACCEPT loop.

Maximize Box

To place a maximize button in your window, mark the **Maximize Box** check box. In general, you should place a maximize button only on application windows and MDI child document windows. This adds the MAX attribute to the window.

Scroll Bars

To add a horizontal scroll bar to your window, mark the **Horizontal** check box. Scroll bars only appear when something inside the window--a control--is bigger than the window. To add a vertical scroll bar to your window, check the **Vertical** check box. These options add the HSCROLL, VSCROLL, and HVSCROLL attributes to the window.

Assuming your application frame will display MDI child windows, you normally check both **Horizontal** and **Vertical**.

Status Widths

To set the width of the status bar zone(s), type a value or list of values in the **Status Widths** field. You must also check the **Status Bar** box in the top part of the dialog to display a status bar. The values you enter in this field fill the STATUS attribute parameters.

The zones are the areas within the status bar marked off by the 3D shaded boxes. The first zone on the left, by default, displays MSG attribute text. This is useful for specifying short help instructions or other information to the user. If your application has only one zone for the status bar, you may omit this field. For more than one zone, enter a series of comma-separated values. The default measurement unit is dialog units.

You may set a minimum value for a zone width by typing a negative number. This creates a zone with a minimum width, but is expandable by resizing the window. Use the runtime property assignment syntax to place text in any zone. To place a string in the second zone, for example:

```
{PROP:StatusText} = ?array
```

Tip

A multi-zone status bar can give your application a professional look. You may display help text in zone one, and when editing a record, the current record number in zone two, for example.

Position

Lets you set the location and size of a window.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the window's **Top Left Corner**, choose from the following options for the **X** (horizontal) and **Y** (vertical) coordinates.

Default	The Windows operating system determines the initial position of the window. By default, Windows position windows in a cascading sequence from top-left to bottom-right.
Center	Centers an APPLICATION window on the screen. Centers child windows on their respective parents.
Fixed	Lets you set a precise coordinate in Dialog units. Generally, the coordinate is relative to the top left corner of the screen for APPLICATION (first or main) windows and relative to the top left corner of the APPLICATION window for all subsequent (child) windows.

To set the window's size, choose from the following options for the **Width** and **Height**.

Default	The Windows operating system determines the initial size of the window.
Fixed	Lets you set a precise width or height in Dialog units. Dialog units provide a relative screen measure based on the system font character size.

Actions

Tables	Accesses the **Table Schematic Definition** dialog for this procedure.
Embeds	Accesses the **Embedded Source** dialog for points surrounding the event handling for this procedure.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for the selected control or window. As you choose options, the dialog box displays a sample of the selected font.

List Box Formatter Dialog

The List Box Formatter dialog shows how the list box under construction looks. It fills this sample list box with placeholder characters representing the contents of each field. If any field contains a header, a header row appears over the column.

You format the fields one by one in the **List Field Properties**. The sample list box always displays a horizontal scroll bar, whether you specify one in the **List Properties** dialog or not.

The formatter does *not* display a vertical scroll bar. If the queue contains more items than rows in the list, and if you add the VSCROLL attribute by checking the box in the **List Properties** dialog, the vertical scroll bar appears at run time.

The dialog contains the following buttons:

New Column	To add a field to the list box, press the **New Column** button (or INS). When opening the **List Box Formatter** from within the Application Generator, using a procedure template which supports it, the **Populate** button displays the **Select Field** dialog. From here, you can indicate any database field or memory variable for use as a list box column. The generated code puts the contents of the database records into the queue for use in the list box.
New Group	To add a new Group, press the **New Group** button (or SHIFT+INS). This adds the currently highlighted column into the new group. Use the **Move Column** buttons to move other columns into (or out of) a group.
Delete	Removes the currently selected field from the list box.
(Arrow Left)	Moves the currently selected field one position to its left. If the selected field is the leftmost in a group, the field moves out of the group, without changing position. If the selected field is immediately to the right of a group, the field moves into the group, without changing position.
(Arrow Right)	Moves the currently selected field one position to its right. If the selected field is the rightmost in a group, the field moves out of the group, without changing position. If the selected field is immediately to the left of a group, the field moves into the group, without changing position.

See the following topics in the *FAQ* PDF for more information about adding list box functionality:

How to Restore User Resized List Box Column Widths

How to Trap a Double Click on a List Box

How to add Drag and Drop to a List Box

How to Display the Sort Field First on a Multi-Key Browse

Using drop-down lists to Lookup Records

General

The tab lets you set the following formatting options.

Column Heading

Heading Text

Optionally specify header text for the column. The header appears as a gray row above the list box data items. To specify no header, leave this field blank. If any field included in the list box has a header, a header appears across the entire list box; fields with no header text have a blank header.

The heading appears within the FORMAT string enclosed in tilde (~) characters, as in "~My Header~."

Width

Specify the width in dialog units for the column data. By default, the Formatter sets the value to four times the number of characters specified in the field picture in the data dictionary. For variables, the default is four times the number of characters in the picture token defined for it.

Tip: As a rough guide, allow four dialog units for an average character. For example, if you want a column 10 characters wide, type 40 in the Width field.

Tip: After you've placed a field, you can drag the column separators in the Sample list box to resize the column width. The cursor changes when you place it on top of the separator, to indicate you can resize it.

The data width you set appears within the FORMAT string for the field, preceding the Justification code, as in "40L."

Scroll Bar

Type a non-zero value to specify a horizontal scroll bar for this column only. If the overall list box already has a scroll bar, the column scroll bar appears above the list box scroll bar. The value specifies, in dialog units, how far the column scrolls.

For example, if your *data* is fifty (50) characters, and your list box *column* width is about forty (40) characters (one hundred sixty (160) *dialog units*), you should specify a value of fifty (50). Fifty (50) additional dialog units are enough to display the ten characters that extend beyond the width of the list box column.

The scroll bar and size appear in the FORMAT string together, as in "S(4)."

Justification

Choose from the drop-down list to specify left, right, center or decimal header justification.

This appears within the FORMAT string following the header, as in "~My Header~L."

Indent

Optionally specify an indent, in dialog units, for the heading text. Indent moves the data by the number of dialog units specified, in the opposite direction to the justification. An indent of two (2) on left justification improves listbox readability.

This appears within the FORMAT string following the header, as in "~My Header~L(8)."

Column Data

Picture

Specify the picture token for the data. The **List box Formatter** displays the data according to the picture token. For example, the picture token @P(###) ###-####P displays a phone number as (555) 555-5555.

The picture token you specify appears in the FORMAT string.

Justification

Choose from the drop-down list to specify left, right, center or decimal. Decimal justification aligns decimal numbers by their decimal points.

The justification appears in the FORMAT string following the data width, as in "40R."

Indent

Optionally specify an indent, in dialog units, for the listbox data. Indent moves the data by the number of dialog units specified, in the opposite direction to the justification. An indent of two (2) on left justified data improves listbox readability.

The indent appears within the FORMAT string surrounded by parentheses and preceded by a letter indicating the justification, as in "L(8)."

Field Number/Auto

The General tab now includes support for the Field Number of a column, and the ability to override it. The Auto check box designates that the field numbering of the selected column will be "standard", based on the order that it appears in the list box. Turning the Auto attribute off allows the field numbering to be modified. This feature is important for some list box properties that require field number information, and gives you the ability to control it.

Formatting Flags

Right Border

Check this box to specify column separators between fields in the list box at run time.

The FORMAT string includes the pipe symbol (|), immediately preceding the header text, as in "|~MyHeader~."

Resizeable

Check this box to specify that the user can resize the width of the columns at run time.

The FORMAT string includes the "M" character, immediately preceding the header text as in "M~MyHeader~."

See How to Restore User Resized List Box Column Widths...

Underline

Check this box to add the underline style to the list box text. In effect, this creates a bottom border for each row in the column, giving your list box a spreadsheet or cell-like appearance.

The FORMAT string includes the underscore character, immediately preceding the header text, as in "_~My Header~."

Fixed

Check this box to specify that the column always remains visible in the list box, even if other columns scroll.

The FORMAT string includes the "F" character, immediately preceding the header text as in "F~MyHeader~."

Last on Line

Checking this box specifies that the next field in the group will appear immediately below the current field. In effect, it stacks two or more fields below the group header.

Tip: The field must be part of a group. See *Creating Column Groups*.

The FORMAT string includes the "/" character, immediately preceding the header text as in "/~MyHeader~."

Locator

By default, the first field in a multi-column COMBO displays in the entry portion of the COMBO. Check the **Locator** box to specify that this field (instead of the first field) displays in the entry box portion of a multi-column COMBO control.

The FORMAT string includes the "?" character, immediately preceding the header text as in "?~MyHeader~."

Color

Check this box to allow conditional runtime colors for individual list items--that is, to conditionally override the default colors for individual list rows. The color information for each row is contained in four LONG fields that immediately follow the data field in the QUEUE. Assign the color value to the appropriate QUEUE field at runtime, and Clarion's runtime library does the rest.

See *Control Templates--BrowseBox Control* in the *Template Guide* for information on specifying conditional BrowseBox colors, and see *FORMAT* in the *Language Reference*.

Adds an asterisk "*" to the FORMAT string.

Style

Check this box to allow conditional runtime fonts for individual list items--that is, to conditionally override the default fonts for individual list rows. The font (style) information for each row is contained in a LONG field that immediately follows the data field in the QUEUE. Assign the style value

to the appropriate QUEUE field at runtime, and Clarion's runtime library does the rest.

See *Control Templates--BrowseBox Control* in the *Template Guide* for information on specifying conditional BrowseBox colors, and see FORMAT in the *Language Reference*.

Adds a "Y" to the FORMAT string.

Tooltip

Check this box to activate a specific tool tip for the selected column. The tool tip content is specified on the Appearance tab.

Adds a "P" to the FORMAT string.

Tip: Before you can activate individual column tool tips, make sure that you have a tool tip defined for the list box control. This is found on the Help tab of the selected list box control.

Appearance Tab

Color and Style

Use these prompts to set the default colors for all list rows and columns.

Text

To set the default color for normal (unselected) list text, type a valid color equate in this field, or press the ellipsis (...) button to select a color from the color dialog.

Adds an "E(*color*,,,)" to the FORMAT string.

Background

To set the default color for normal (unselected) list background, type a valid color equate in this field, or press the ellipsis (...) button to select a color from the color dialog.

Adds an "E(,*color*,,)" to the FORMAT string.

Selected Text

To set the default color for normal (unselected) list text, type a valid color equate in this field, or press the ellipsis (...) button to select a color from the color dialog.

Adds an "E(,,*color*,)" to the FORMAT string.

Selected Background

To set the default color for normal (unselected) list background, type a valid color equate in this field, or press the ellipsis (...) button to select a color from the color dialog.

Adds an "E(,,,*color*)" to the FORMAT string.

Default Style

Type the default style number. The style number sets the font typeface, size, style, and color for all list rows and columns.

Adds a "Z(*n*)" to the FORMAT string, where *n* is the style number.

Icon

None

Select this to display no icons in the column.

Normal

Select this to create an area to the left of the data in the column for displaying a normal image (.ICO) that you supply. See *Control Templates--BrowseBox Control* in the *Template Guide* for information on specifying BrowseBox icons, and see *Prop:IconList* in the *Language Reference*.

Adds an "I" to the FORMAT string.

Transparent

Select this to create an area to the left of the data in the column for displaying a transparent image (.ICO) that you supply. See *Control Templates--BrowseBox Control* in the *Template Guide* for information on specifying BrowseBox icons, and see *Prop:IconList* in the *Language Reference*.

Adds a "J" to the FORMAT string.

Tree

Tree

Checking this box displays this column in a hierarchical tree diagram. See Relation Tree control template.

Adds a "T" to the FORMAT string.

One-based tree

Checking this box allows the root level to collapse, that is, all the items in the tree can collapse to a single line.

Adds a "(1)" to the "T" in the FORMAT string, resulting in "T(1)."

Show Level

Checking this box causes each descending level of the Tree hierarchy to be indented.

Clearing this box appends "(I)" to the "T" in the FORMAT string, resulting in "T(I)."

Show Lines

Checking this box adds connecting lines between related items in the tree diagram.

Clearing this box appends "(L)" to the "T" in the FORMAT string, resulting in "T(L)" to suppress lines.

Show Boxes

Checking this box adds expand (+) and contract (-) boxes to the tree diagram.

Clearing this box appends "(B)" to the "T" in the FORMAT string, resulting in "T(B)" to suppress boxes.

Show Root

Checking this box displays a root item for the tree diagram.

Clearing this box appends "(R)" to the "T" in the FORMAT string, resulting in "T(R)" to suppress display of a root item.

Default Column Tip

Used to designate default text to be used for the selected column's tool tip.

Adds a "Q" to the FORMAT string.

Tip

It is not necessary to be concerned with the precise syntax of the List Box Format String. Always use the List Box Formatter to build the string for you. If you are using PROP:FORMAT in your embedded source, you can always cut and paste your format string from the Formatter as needed.

List Box Styles

A list box style is a collection of fonts and colors that control a list box appearance. Applying a style to a list box is a two-step process. The style attribute must first be enabled for each desired column in the List Box Formatter. Next, a style can be applied in the Actions tab for each column in the list box control. The styles that are applied are maintained and organized in this window.

This dialog provides update buttons to add, change or delete a selected list box style.

In the subsequent **Styles** dialog, the following prompts are provided:

Note:

Each prompt that follows in this dialog has an optional "V" button that, when pressed, provides an ellipsis button for variable selection. This extends the power of list box styles to allow a given style to be controlled and modified at runtime.

Style Attributes

A list box style contains the following attributes:

Font
Press the **Font** button to designate a **Font Name** (Arial, MS Sans Serif, etc.), **Font Size**, and **Font Style** (Regular, Bold, Italic, etc.) for the active list box style. Manual entry of these settings is also permitted, but entering incorrect values (Font Names and Styles that are not supported) can produce unpredictable results.

Color
Press the appropriate ellipsis button to set the active style colors for a column's **Foreground Normal**, **Foreground Selected**, **Background Normal**, and **Background Selected** display mode.

Picture

Press the ellipsis button to build a picture token, or enter a picture token in the entry field provided which will replace the default column picture.

Runtime List Box

The **Runtime List Box** style option is used to generate the appropriate runtime property syntax for a selected style and a selected list box that is not associated with the default procedure template (i.e., a hand coded list box or a third party template that populates a list box). Select the list box to associate with the current active style.

Menu Editor

The **Menu Editor** dialog visually represents a Clarion MENUBAR data structure. The menu tree (on the left hand side of the dialog) appears as simplified Clarion language syntax, containing these Clarion keywords:

A MENUBAR keyword at the top.

A MENU statement or statements followed by a menu name, and a corresponding END statement.

An ITEM statement or statements followed by an item name.

Menu Editor command buttons allow you to add and delete MENUs and ITEMs. You may also move MENUs and ITEMs within the MENUBAR structure with the ↑ and ↓ buttons.

The right hand side of the dialog lets you specify the text of your MENUs and ITEMs, the equate labels used to reference the MENUs and ITEMs in executable code, and the actions that occur when the user selects an ITEM.

Tip

When using the Application Generator, each ITEM you place on a MENU or MENUBAR automatically adds an embed point to the control event handling tree in the Embedded Source dialog. This lets you easily attach functionality to your ITEMs.

Menu Editor Buttons

New Menu

This button adds a new MENU statement, its Menu Text, and its corresponding END statement. The MENU is added after the highlighted line. MENUs may be nested within other menus. MENUs may contain MENUs or ITEMs.

New Item

This button inserts an ITEM after the highlighted line. Note that ITEMs are used to execute commands or procedures, whereas MENUs are used to display a selection of other MENUs or ITEMs.

New Separator 

To add a separator bar after the currently highlighted MENU or ITEM, press the **Separator button**.

Separator bars can provide the user with a visual cue that a group of ITEMs on the menu perform related functions.

Delete Menu/Item Button 

To delete the currently highlighted MENU, ITEM, or SEPARATOR, press the **Delete** button. If you delete a MENU statement, all ITEMs and MENUs within it, and its associated END statement are also deleted.

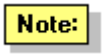
- and ¯ **Buttons** 

To move the currently highlighted MENU, ITEM, or SEPARATOR up or down in the menu list, press the **-** or ¯ button. When moving a MENU, all ITEMs and MENUs within it, and its associated END statement move also.

General

Menu Text

Type the text you want displayed for this MENU or ITEM. For example, type &FILE, so the end user sees **File**. The ampersand within the Menu Text field signifies the character following the ampersand is the accelerator key. That is, the character is underlined, and, when the user presses the accelerator key, the action associated with the ITEM is executed.



A MENU accelerator key requires THE ALT key to take effect, whereas an ITEM accelerator key does not require the ALT key, but does require that the ITEM be currently displayed. See Hot Key below for another method of accessing your MENUs and ITEMs.

Use Variable

This defines the USE attribute for the control. Type a Field Equate Label.

A Field Equate Label has a leading question mark (?), and you should make it descriptive. For example ?File shows this menu is to manipulate a file. You can refer to the MENU within executable code by its Field Equate Label.

Message Type the MSG attribute contents.

This message text displays in the status bar (if enabled) when the user highlights this MENU or ITEM.

Help ID Type either a help keyword or a context string present in a .HLP file.

If you fill in the **Help ID** for a MENU or an ITEM, when the user highlights the MENU or ITEM and presses F1, the help file opens to the referenced topic.

The **Help ID** field (HLP attribute) takes a string constant specifying the key for accessing a specific topic in a Windows Help file. This may be either a Help keyword or a context string. When referencing a context string in the **Help ID** field, you must identify it with a leading tilde (~).

STD ID To specify a standard windows action for your menu ITEM, enter one of the equates listed below in the **Std ID** field. Clarion will automatically implement the command using standard windows behavior; you do not need any other support for it in your code. The standard equate labels and their associated actions are also contained in the ..\LIBSRC\EQUATES.CLW file.

STD:PrintSetup	Printer Options Dialog.
STD:Close	Closes active window.
STD:Undo	Reverses the last editing action.
STD:Cut	Deletes selection, copies to clipboard.
STD:Copy	Copies selection to clipboard.
STD:Paste	Pastes clipboard contents at the insertion point.
STD:Clear	Deletes selection.
STD:TileWindow	Arranges child windows edge to edge.
STD:TileHorizontal	Arranges child windows edge to edge.
STD:TileVertical	Arranges child windows edge to edge.
STD:CascadeWindow	Arranges child windows so all title bars are visible.
STD:ArrangeIcons	Arranges iconized child windows.
STD:WindowList	Adds child window names to menu.
STD:Help	Opens .HLP file to the contents page.
STD:HelpIndex	Opens .HLP file to the index.
STD:HelpOnHelp	Opens Microsoft's .HLP file for the Windows Help system.
STD:HelpSearch	Opens Microsoft's Help Search utility for the .HLP file.

Position Lets you specify MENU and ITEM order priority when Clarion merges menus. The choices are:

To allow normal ordering when merging menus, choose ***Normal*** from the **Position** drop down list. In normal merging, Global selections precede Local selections. See *Merging Menus* in the *User's Guide*.

To force the selected MENU or ITEM to the first position when merging menus, choose ***First*** from the **Position** drop down list. This adds the FIRST attribute to the MENU or ITEM statement.

To force the menu or item to the last position when merging menus, choose **Last** from the **Position** drop down list. This adds the LAST attribute to the MENU or ITEM statement.

Hot Key

Press this button to open the **Input Key** dialog. Use this dialog to add the KEY attribute to your MENU or ITEM. The KEY attribute specifies a "hot" key or key combination.

A hot key is very similar to an accelerator key. A hot key or hot key combination allows the end user to immediately display a MENU, or execute the action associated with an ITEM, without mouse clicking, and without displaying the menu that contains the ITEM. Customarily, hot keys take the form of CTRL + *character*, or CTRL + SHIFT + *character*.

Tip

You may want to add the hot key combination to the menu text to signal its availability to the user. See the *Windows Design* appendix in the *User's Guide* for a list of common hot keys associated with standard windows commands.

Menu Text

Type the text you want displayed for the Hot Key. This text is right justified on the menu.

Flags

Disable Item

To disable a MENU or ITEM (dim the text and make it unavailable to the user), check the **Disable Item** box. This adds the DISABLE attribute to the MENU or ITEM statement.

Tip

The Disable box is handy when you incorporate modality into a program--that is, when one type of child window does *not* support the same commands another type does. For the type that doesn't support the command, disable the ITEM rather than omitting it. This will avoid confusing the user with menu ITEMs that disappear and reappear depending on which window is active.

Toggle (on/off) Item To create an on/off toggle for a selected ITEM, check the **Toggle (on/off) Item** box. The ITEM should have a numeric variable in the **Use Variable** field. The variable should be declared using one of the data dialogs, or in embedded source. The **Menu Editor** adds the CHECK attribute to this ITEM.

With the CHECK attribute, when the user selects the ITEM for the first time, the ITEM is "on," the Use Variable's value is one (1), and a check mark appears beside the ITEM. When the user selects the ITEM a second time, the ITEM is "off," the Use Variable's value is zero (0) and no check mark is displayed. You should add source code to control the application's behavior depending on the state of the Use Variable.

Right Justify To right justify the selected MENU on the action bar, check the **Right Justify** box. This is available only for MENUs on the action bar. Nested MENUs (subMENUs) cannot be right justified. Checking this box displays the selected MENU, and all MENUs after the selected MENU, at the far right of the action bar.

Do Not Merge To tell Clarion never to merge this MENUBAR with other MENUBARs, check the **Do Not Merge** box. This is available only for the MENUBAR, not for MENUs or ITEMs. See the *Language Reference* for more information on the NOMERGE attribute.

Appearance

Icon Specify the ICON you want displayed for this MENU or ITEM.

Colors Enter a valid color equate in the Text Color or Background fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your MENU or ITEM declaration.

Left Offset Specify the level of indention for the MENU or ITEM (in dialog units). This sets the parameter for the LEFT attribute.

Width and **Height**.

Default The Clarion runtime library determines the size based on the applicable font and text value.

Fixed Lets you set a precise width or height in Dialog units on a window.

Font Calls the **Select Font** dialog which lets you change the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for the selected MENU or ITEM As you choose options, the dialog box displays a sample of the selected font.

Actions

Use the Actions tab to add functionality to your menu item. Filling in these prompts causes the menu item to execute an action when the user selects the menu item.

When Pressed From the drop down list, choose *Call a Procedure, Run a Program*, or *No Special Action*.

The procedure or program you specify executes when the user selects the menu item. The choices are:

Call a Procedure You must specify the **Procedure Name**, and whether the procedure will **Initiate a Thread**.

Procedure Name From the **Procedure Name** drop down list, choose an existing procedure name, or type a new procedure name. A new procedure appears as a "ToDo" item in your Application Tree.

Initiate a Thread Optionally check the **Initiate a Thread** box. If the procedure initiates a thread, specify the Thread Stack size. Clarion uses the START function to initiate a new execution thread. You can optionally specify **Parameters**, **Requested File Action**, or both.

Tip

A MENU ITEM on an application frame toolbar that calls an MDI child procedure <u>must</u> initiate a thread.

Thread Stack Accept the default value in the **Thread Stack** spin box unless you have extraordinary program requirements. To change the value, type in a new value or click on the spin box arrows.

Parameters In the **Parameters** field, optionally type a list of variables or data structures passed to the procedure.

Requested File Action

From the drop down list, optionally select ***None, Insert, Change, Delete***, or ***Select***. The default selection is ***None***. The Global Request variable gets the selected value. The called procedure can then check the value of the Global Request variable and perform the requested file action.

Run a Program You must specify the **Program Name**, and optionally, any parameters.

Program Name Type the program name.

Parameters Optionally type a list of values that are passed to the program.

No Special Action Choose this option if you are providing your menu item's functionality with another method, such as embedded source, or an STD ID.

Note:

You may combine a procedure or program call with embedded source, but not with an STD ID.

Tables Accesses the **Table Schematic Definition** dialog for this procedure.

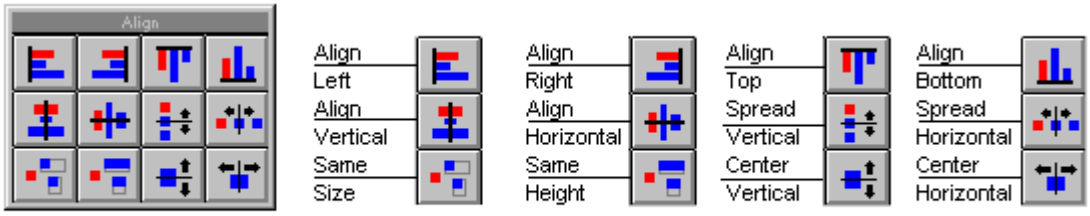
Embeds Accesses the **Embedded Source** dialog for points surrounding the event handling for this menu item only.

Using the Window Formatter - Align Toolbox

The **Window Formatter's Align** toolbox allows you to quickly, professionally, and precisely align the controls in your window.

Display or hide the **Align** toolbox by choosing **Options ▸ Alignbox.** Resize the **Align** toolbox by placing the cursor on the border of the box. When the cursor changes to a double-headed arrow, CLICK and DRAG.

Select the controls to align (CTRL+CLICK allows you to select multiple controls, or you can "lasso" multiple controls with CTRL+DRAG), then click on the appropriate alignment tool. All the alignment actions are also available from the **Alignment** menu.



Tip

For most alignment functions, the first controls selected (blue handles) are aligned with the last control selected (red handles). That is, the last control selected is the anchor control. It doesn't move, the others do.

Tip

Position the cursor over any tool and wait for half a second. A tool tip appears telling you the type of alignment this tool will accomplish.

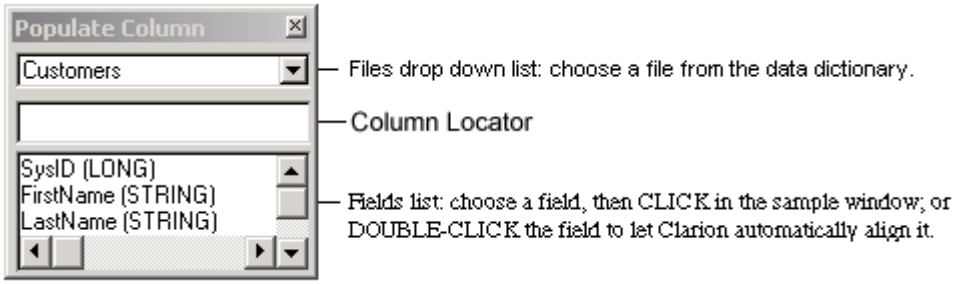
Align Left	Aligns the left borders of the selected controls with the left border of the last control selected (red handles).
Align Right	Aligns the right borders of the selected controls with the right border of the last control selected (red handles).

Align Top	Aligns the top borders of the selected controls with the top border of the last control selected (red handles).
Align Bottom	Aligns the bottom borders of the selected controls with the bottom border of the last control selected (red handles).
Align Vertically	Along a vertical axis, aligns the centers of the selected controls with the center of the last control selected (red handles).
Align Horizontally	Along a horizontal axis, aligns the centers of the selected controls with the center of the last control selected (red handles).
Spread Vertical	Equalizes the vertical spaces between the selected controls.
Spread Horizontal	Equalizes the horizontal spaces between the selected controls.
Same Size	Makes all selected controls the same height and width as the last control selected (red handles).
Same Height	Makes all selected controls the same height as the last control selected (red handles).
Center Vertically	As a group (relative positions of selected controls don't change), centers the selected controls horizontally within the window.
Center Horizontally	As a group (relative positions of selected controls don't change), centers the selected controls vertically within the window.

Using the Window Formatter - Column Toolbox

The **Window Formatter** contains a floating **Populate Column** toolbox. This toolbox allows you to quickly "populate" a window with entry controls **and** prompts for fields in your data dictionary tables.

Display or hide the **Populate Column** toolbox by choosing **Options ▸ Fieldbox.** Resize the **Populate Column** toolbox by placing the cursor on the border of the box. When the cursor changes to a double-headed arrow, CLICK and DRAG.



1. Choose a **table** from the drop down list.
2. Select the **column** you want on your window.
3. DOUBLE-CLICK the column to let Clarion automatically align the controls, or just CLICK on the column then CLICK in the sample window to place the control **and** its associated prompt.

   The **type** of control (entry box, check box, radio button, etc.) is determined by the settings for this particular column in the Data Dictionary.

Using the Window Formatter - Command Toolbox

The **Window Formatter** contains a dockable **Command** toolbox. The toolbox lets you quickly execute a variety of **Window Formatter** functions at the touch of a button.

All the commands in the **Command** toolbox are also available from the menu (**Exit!, Edit, View, Preview!**).

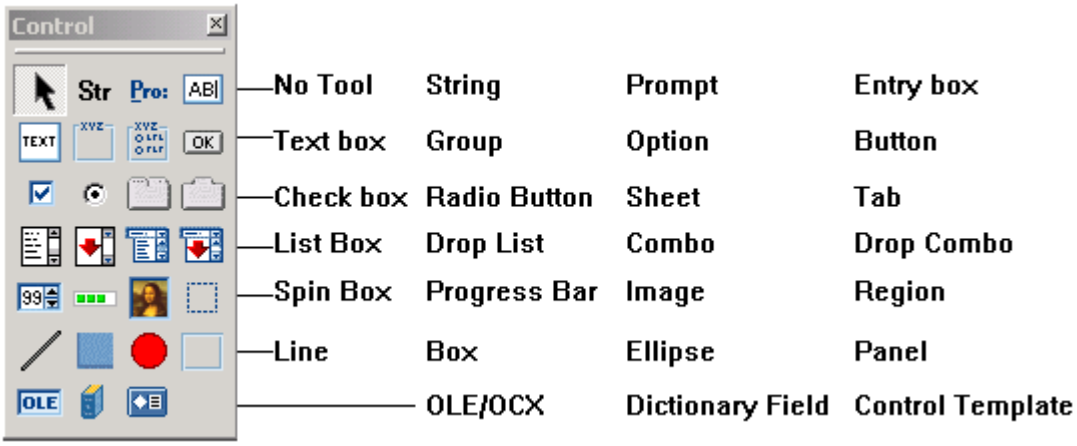
Display or hide the **Command** toolbox by choosing **View ▸ Show Commandbox.** Resize the **Command** toolbox by placing the cursor on the border of the box. When the cursor changes to a double headed arrow, CLICK and DRAG. Dock the toolbox by dragging the handle (double vertical lines) to any edge of the Window Formatter frame (dragging the title bar repositions the toolbox *without* docking).

Tip: Position the cursor over any tool and wait for half a second. A tool tip shows you the type of control this tool creates.

Save Changes	Exit the Window Formatter and save changes.
Lose Changes	Exit the Window Formatter and abandon changes.
Properties	Edit the properties of the selected (red handles) control or window.
Actions	Edit the actions of the selected (red handles) control or window.
Embeds	Edit the embedded source for the selected (red handles) control or window.
Show Toolbox	Hide or display the Controls Toolbox.
Show Alignbox	Hide or display the Align Toolbox.
Show Property Box	Hide or display the Property Toolbox.
Show Field Box	Hide or display the Fields Toolbox.
Grid	Toggle the alignment grid on or off.
Preview	Preview the window.

Using the Window Formatter - Controls Toolbox

The **Window Formatter** contains a floating **Controls** toolbox, similar to those found in many draw or paintbrush programs. Simply choose a control from the toolbox (CLICK on it), then CLICK in the sample window to place the control in the window.



Display or hide the **Controls** toolbox by choosing **Options ▸ Toolbox.** Resize the **Controls** toolbox by placing the cursor on the border of the box. When the cursor changes to a double headed arrow, CLICK and DRAG. All the controls in the toolbox are also available from the **Controls** menu.

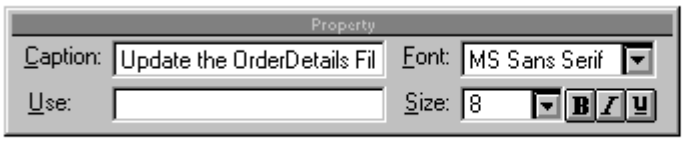
Tip

Position the cursor over any tool and wait for half a second. A tool tip appears telling you the type of control that will be created by this tool.

Using the Window Formatter - Property Toolbox

The **Window Formatter's Property** toolbox allows you to quickly specify the appearance and content of the text on each control within the window and on the window title bar. Control the font, size, style, and content of all your text, using standard word processor buttons and drop down lists.

Display or hide the **Property** toolbox by choosing **Options ▸ Propertybox.** Resize the **Property** toolbox by placing the cursor on the border of the box. When the cursor changes to a double-headed arrow, CLICK and DRAG.



Text Formatting: choose content, font, size, and style for the selected control or window.

Using the Window Formatter - Sample Window

The **Window Formatter** is a visual design tool. You always see a sample of the window you're working on, as you work on it. For example, place a list box in the sample window and drag its handles to the size you want.

In addition, you can see the window, exactly as it will appear to the end user by choosing **Preview!** from the action bar.

Window Formatter Options

The **Window Formatter Options** dialog sets the default position and size values applied when auto-populating controls, or when aligning controls with the alignment tools. To access the dialog, choose **Setup ▸ Window Formatter Options** from the environment menu, or choose **View ▸ Options** from the Window Formatter menu. The dialog is divided into four sections or tabs: Grid, Populate Defaults, Margin Defaults, and Spread Defaults.

Grid

This tab turns grid snap on or off, and sets the starting point and offsets of the window grid. It also lets you show or hide the screen boundaries (extents) for the most common video resolutions (640x480, 800x600, 1024x768, etc.).

You can use the grid to force the boundaries of your window controls to fall only on certain x / y values (axes, latitude, longitude). By enforcing the grid axes, your controls are easier to position and align.

Snap to Grid

Check this box turn grid snap on; clear the box to turn it off. Grid snap displays a dot grid of valid positioning coordinates and forces the upper left corner of new controls to align with the dot grid. The end user does not see the grid at run time; it is a design tool only.

Tip: You can also choose View ▸ Show Grid from the menu, or press the button to toggle grid snap on and off.

Width

Enter the horizontal distance between the grid dots (x axis). This is the minimum horizontal distance you can move a control when grid snap is on.

Height

Enter the vertical distance between the grid dots (y axis). This is the minimum vertical distance you can move a control when grid snap is on.

Origin X

Enter the horizontal coordinate at which to begin placing the grid dots. This is the left-most position at which controls will align or auto-populate when grid snap is on.

Origin Y

Enter the vertical coordinate at which to begin placing the grid dots. This is the top-most position at which controls will align or auto-populate when grid snap is on.

Tip: All grid values are specified in dialog units, a unit of measure based on the current system font. See the Glossary for a complete definition.

Show Screen Extents

Check this box to show video screen boundaries within the Window Formatter for the most common video resolutions. Clear the box to suppress the boundaries.

Snap on Resize

Check this box to force controls to snap to the nearest grid point grid when resizing from the right or bottom edges. This constrains a controls width and height to the grid. Resizing a control using the top and left edges always snaps to the grid.

Populate

This tab sets the default width and height for a variety of window controls. The Window Formatter applies the default sizes whenever you use it to add a control to the window.

Control Type

Choose the type of control for which to set the default size.

Tip: The default sizes are specified in dialog units, a unit of measure based on the current system font. See the Glossary for a complete definition.

Default Width

Set the default width for the specified control type. A value of zero (0) specifies no width, the control expands to the size of the data it displays. See *AT* in the *Language Reference* for more information.

Use most common width already present

Check this box to specify a dynamic default based on width of any controls of the same type that are already present on the window. For example, if there are three ENTRY controls and two of the controls are 50 units wide, then 50 becomes the default width for ENTRY controls. Clear this box to always apply the Default Width value, even if other controls of the same type are present.

Default Height

Set the default height for the specified control type. A value of zero (0) specifies no height; the control expands to the size of the data it displays. See *AT* in the *Language Reference* for more information.

Use most common height already present

Check this box to specify a dynamic default based on height of any controls of the same type that are already present on the window. For example, if there are three ENTRY controls and two of the controls are 10 units wide, then 10 becomes the default height for ENTRY controls. Clear this box to always apply the Default Height value, even if other controls of the same type are present.

Margins

This tab sets the margins applied by the margin alignment tool. For more information on these alignment tools, see *Window Formatter Tools*. The margin is simply the distance between the closest edges of two controls (or of a control and the window). The Window Formatter applies the margins whenever you use the margin alignment tools.

Different types of controls require different margins to accommodate their unique characteristics. For example, TAB controls and GROUP controls need extra space to allow for their text.

Container Type

Choose the type of control for which to set the margins. Choose from:

Window	Set the default margins.
Group	Set the margins to apply for controls inside a GROUP control, abutting the GROUP control.
Option	Set the margins to apply for controls inside an OPTION control, abutting the OPTION control.
Tab	Set the margins to apply for controls inside a TAB control, abutting the TAB control.

Top Margin

The distance between the top edge of the selected control and the nearest horizontal edge of a bounding control or window.

Side Margins

The distance between the vertical edges of the selected control and the nearest vertical edge of a bounding control or window.

Bottom Margin

The distance between the bottom edge of the selected control and the nearest horizontal edge of a bounding control or window.

Tip: The two settings (Full and Thin) provide alternative margins applied by the margin alignment tools. For more information on these alignment tools, see *Window Formatter Tools*.

Tip: The default sizes are specified in dialog units; a unit of measure based on the current system font. See the Glossary for a complete definition.

Spread

This tab sets the default spacing between auto-populated window controls and between controls positioned by the Spread Alignment tools. The Window Formatter applies the default spacing when you auto-populate fields from the Fields Toolbox and when you use the Spread Alignment tools. For more information on these alignment tools, see *Window Formatter Tools*.

Button Spacing

Set the default distance between the edges of a button control and the nearest control.

Other Control Spacing

Set the default distance between the edges of two adjacent controls.

Prompt to Field Spacing

Set the default distance between the right edge of a PROMPT control and the left edge of its (visually) associated (ENTRY, SPIN, TEXT, etc.) control.

Tip: The default sizes are specified in dialog units, a unit of measure based on the current system font. See the Glossary for a complete definition.

Multiple control alignments

This selection applies only to the Spread Alignment tools. For more information on these alignment tools, see *Window Formatter Tools*.

Use fixed spacing

Apply the static values specified above when spacing (spreading) multiple controls.

Calculate spacing

Calculate spacing based on distance between first and last control, so there is an equal distance between each control.

Window Properties

This dialog lets you specify the appearance and functionality of your window.

General

Text

To specify caption bar text, type a string constant in the **Text** field. The caption bar holds the name of the window.

Tip

You may dynamically alter the caption bar text at run-time. See PROP:Text in the Language Reference.

Label

This names the specific WINDOW in the source code. The label may contain upper or lower case letters, numerals, the underscore character or a colon. Space characters are forbidden. The first character must be a letter or the underscore character. Clarion reserved words may not serve as labels.

Frame Type

To choose the frame for your window, pick a selection from the **Frame Type** drop-down list. The frame defines the borders of the window. Choose from:

Single - a single pixel frame which the user cannot resize. Most suitable for dialog boxes.

Double - a thick frame, which the user cannot resize. Use this type frame for a system modal window with no caption bar, or for a modal dialog box with a caption bar. This adds the DOUBLE attribute to the window.

Resizeable - a thick frame, which the user can resize. Choose this for application and MDI child windows. This adds the RESIZE attribute to the window.

Layout

Indicates the orientation of window controls and field sequence.

Left to Right maintains the original layout specified in the Window Formatter.

Default field navigation moves from left to right.

Right to Left essentially "flips" the window controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child window that have the **Default** setting active.

Initial Size

Sets the initial state of your window. Choose from:

Normal: - displays the window at the default size which either you specifically set, or Windows sets if you don't.

Maximized: - the window fills the desktop, if an application window, or the window frame, if an MDI child window. This adds the MAXIMIZE attribute to the window.

Iconized: - the window appears in iconized state--as a 32 by 32 pixel window at the bottom of the desktop, for an application window, or at the inside bottom of the application frame, for an MDI child window. This adds the ICONIZE attribute to the window.

Freeze

"Freezes" all the controls on the window so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Wallpaper

To provide a background image for the window’s client area, specify an image filename. Type the filename or press the ellipsis button (...) to select a file. See WALLPAPER in the *Language Reference*.

Wallpaper Mode

Specify how the window displays the background image. Choose from:

Stretched	The image expands to fill the entire client area.
Centered	The image displays at its default size and is centered in the window's client area.
Tiled	The image displays at its default size and is repeated so it fills the entire client area.

Color

Enter a valid color equate in the Text Color, Background, Selected text, or Selected fill fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your window declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Icon

To associate an icon with the window, allowing it to be minimized, specify an icon in this field. You may type in a file name or an EQUATE. You may also press the ellipsis button (...), then select an icon file name using the standard **Open File** dialog. The file name or equate you specify becomes the parameter for the ICON attribute.

You should always specify an icon for an application window, and for an MDI child window. Specifying an icon name automatically places a minimize button on the caption bar of your application or MDI child window.

Palette

Use the PALETTE attribute to specify maximum color depth. The PALETTE attribute specifies how many colors you want this window to use when it is the foreground window. For example, 24-bit color would be 16777215. The number you specify becomes the parameter for the PALETTE attribute. Leave this field blank to specify the default for the end user's system.

Timer

To have the window receive Timer Event messages from Windows, fill in the **Timer** field. Specify the timer interval in hundredths of seconds. The value you specify becomes the parameter for the TIMER attribute.

For example, if you specify 100 in the field, the window will automatically receive an EVENT:Timer once every second (100/100's seconds). This might be appropriate for adding a clock to a status bar.

Immediate

To generate a message event each time the end user moves or resizes the window, check the **Immediate** box. This adds the IMM attribute to the window. You are responsible for the code that executes upon notification of the event.

Status Bar

To provide a message bar at the bottom of your window, check the **Status Bar** box. This adds the STATUS attribute to the window.

Note:

A status bar in an application window is an excellent way to increase user feedback in your application. Clarion makes it simple to post messages on the status bar advising the user of what your application is doing as it does it. Increasing user feedback makes the user feel more in control. This allows the user to feel more confident and be more efficient when using your application.

Modal Window

To specify a system modal window, check the **Modal Window** box. A system modal window prevents all other tasks--even in applications other than your own--from executing until the window is closed. This adds the MODAL attribute to the window.

Entry Patterns

To enable support for an entry mask for controls in the window, check the **Entry Patterns** box. This lets you specify key-in entry patterns for the fields you choose. This adds the MASK attribute to the window.

System Menu

To place a system menu in your window, check the **System Menu** box. Most windows should have a system menu. For users on a system without a mouse, the system menu provides the only means of minimizing, maximizing or re-sizing the window. This adds the SYSTEM attribute to the window.

Tip

Even if you plan that the window should NOT have a system menu when the application is complete, it's good practice to place a system menu on your application while it's under development. By DOUBLE-CLICKING the system menu, or choosing Close, you can close your application should your normal exit procedure fail.

Auto Display

To add the AUTO attribute to your window, check the **Auto Display** box. This automatically updates the contents of all controls on screen through each pass of the ACCEPT loop.

MDI Child

To add the MDI attribute to your window, check the **MDI Child** box. An MDI child window cannot move outside the main application window. A typical use of an MDI window might be to present a different arrangement of the data in your application's database.

Maximize Box

To place a maximize button in your window, check the **Maximize** box. In general, you should place a maximize button only on application windows and MDI child document windows. This adds the MAX attribute to the window.

3D Look

To provide the gray window background, chiseled control look for your application, check the **3D Look** box. This is clearly a style consideration, but will go a long way in giving your application a professional look. This adds the GRAY attribute to the window.

The gray background is not visible when you design your window with the **Window Formatter**. It is, however, visible in test mode, and when your application runs.

Toolbox

To add the TOOLBOX attribute to your window, check the **Toolbox** box. This makes your window always stay on top.

Dock

The options in this group are enabled only if you have checked on the **Toolbox** attribute above. Click on a location to specify the area that the window is allowed to be docked to within the application frame.

The **Initial State** drop list is used to specify where you would like the window to first appear. The choices in the drop list are directly related to the boxes you check.

Scroll Bars

To add a horizontal scroll bar to your window, check the **Horizontal** box. Scroll bars only appear when something inside the window--a control--is bigger than the window. To add a vertical scroll bar to your window, check the **Vertical** box. These options add the HSCROLL, VSCROLL, and HVSCROLL attributes to the window.

Status Widths

To set the width of the status bar zone(s), type a value or list of values in the **Status Widths** field. You must also check the **Status Bar** box in the top part of the dialog to display a status bar. The values you enter in this field fill the STATUS attribute parameters.

The zones are the areas within the status bar marked off by the 3D shaded boxes. The first zone on the left, by default, displays MSG attribute text. This is useful for specifying short help instructions or other information to the user. If your application has only one zone for the status bar, you may omit this field. For more than one zone, enter a series of comma separated values. The default measurement unit is dialog units.

You may set a minimum value for a zone width by typing a negative number. This creates a zone with a minimum width, but is expandable by resizing the window. Use the runtime property assignment syntax to place text in any zone. To place a string constant in the second zone, for example:

```
{PROP:StatusText,2} = 'Record will be Added'
```

Tip

A multi-zone status bar can give your application a professional look. You may display help text in zone one, and when editing a record, the current record number in zone two, for example.

Drop ID

To specify the type of Drag operations this control will accept, type up to 16 *signatures*, separated by commas. The **Window Formatter** adds the **DROPID** attribute to the control, which indicates the control is a valid target for the drag and drop operations identified by the signatures.

Help

Cursor

The *Cursor* field (the CURSOR attribute) lets you specify an alternate shape for the cursor when the user passes the cursor over the control. The drop-down list provides standard cursor choices such as I-Beam and Crosshair. To select an external cursor file (whose extension must be .CUR), choose **Select File** from the drop-down list, then pick the file using the standard file dialog.

Help ID

The **Help ID** field (the HLP attribute) takes a string constant specifying the key for accessing a specific topic in the Help document. This may be either a Help keyword or a context string.

A Help keyword is a word or phrase indexed so that the user may search for it in the *Help* **Search** dialog. When you fill in the HLP attribute for a button, if the entry box has focus, when the user presses F1, the help file opens to the referenced topic. If more than one topic matches a keyword, the search dialog appears.

When referencing a context string in the **Help ID** field, you must identify it with a leading tilde (~).

Message

The **Message** field (the MSG attribute) lets you specify text to display in the first zone of the status bar when the control has focus.

Position

Lets you set the location and size of a window.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the window's **Top Left Corner**, choose from the following options for the **X** (horizontal) and **Y** (vertical) coordinates.

Default	The Windows operating system determines the initial position of the window. By default, Windows position windows in a cascading sequence from top-left to bottom-right.
Center	Centers an APPLICATION window on the screen. Centers child windows on their respective parents.
Fixed	Lets you set a precise coordinate in Dialog units. Generally, the coordinate is relative to the top left corner of the screen for APPLICATION (first or main) windows and relative to the top left corner of the APPLICATION window for all subsequent (child) windows.

Tip

To give your application the "standard" look of other Windows applications, where possible specify the *Default* setting for any windows with resizable frames.

To set the window's size, choose from the following options for the **Width** and **Height**.

Default	The Windows operating system determines the initial size of the window.
Fixed	Lets you set a precise width or height in Dialog units. Dialog units provide a relative screen measure based on the system font character size.

Tip

Sizing all windows and controls in dialog units lets you design a screen at one resolution, and expect it to look similar at another--in theory. In practice, there can be differences, especially when you display bitmaps in Image controls. Therefore, test your applications in the popular Windows resolutions. The most popular are 640 x 480, 800 x 600, and 1024 x 768 pixels.

Actions

Tables	Accesses the **Table Schematic Definition** dialog for this procedure.
Embeds	Accesses the **Embedded Source** dialog for points surrounding the event handling for this procedure.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for the selected control or window. As you choose options, the dialog box displays a sample of the selected font.

Window and Report Control Properties

Box Control Properties

The box control lets you place a square or rectangle in your window or report. You may fill it with a color, specify a border color, and specify that the borders be rounded. The box control cannot receive focus, nor can it generate events.

General

Use

This defines the USE attribute for the control. Type a field equate label to reference the control in executable code.

Layout

Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Line Width

Set the thickness of the box's border by entering a point value in the **Line Width** spin control. The default is 1 point.

Mode

Hide

Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable

Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control.

Scroll

Specifies whether the control should move with the window when the user scrolls the window. By default, (unchecked), the control does not move with the window. Leave the **Scroll** box unchecked to create a control that stays fixed when the user scrolls the window. The **Window Formatter** places the SCROLL attribute on the control when checked.

Freeze "Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in Border Color or Fill Color fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your control declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Round Lets you specify that the box control should have rounded edges.

Position

Lets you set the location and size of the control.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position and size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the control's **Top Left Corner**, set an **X** (horizontal) and **Y** (vertical) coordinate in Dialog units. The coordinate is relative to the top left corner of the structure containing it, that is, the window or the report band.

To set the control's size, choose from the following options for **Width** and **Height**.

Default The Clarion runtime library determines the size of the control based on the applicable font and picture token.

Full The control is the full height or width of the window or report.

Fixed Lets you set a precise width or height in Dialog units on a window, or in thousandths of inch, points, or millimeters on a report.

Actions

Tables Accesses the **Table Schematic Definition** dialog for this procedure.

Embeds Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

Button Control Properties

A push button (BUTTON) is a rectangular area containing text and/or a picture. When the user presses the button, it should execute a command described by the text or picture.

General

Text The text to display on the button. Place an ampersand (&) before the character to act as the accelerator key for the button--this underlines the character as it appears on the button.

Tip

Microsoft recommends you do *not* place an accelerator key on buttons labeled 'OK,' or 'Cancel.'

Use This defines the USE attribute for the control. The field equate label references the button in program statements.

Justification The **Justification** drop down list is used with button icons.

Default places the icon above any text. ***Right-Justified*** places the icon to the right of any text. ***Left-Justified*** places the icon to the left of any text.

Layout Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Mode

Hide Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable

Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control.

Skip

Instructs the **Window Formatter** to omit the control from the Tab Order. When the user TABS from field to field in the dialog box, Windows will not give the control focus. This is useful for seldom-used data fields. The **Window Formatter** will place the SKIP attribute on the control.

Scroll

Specifies whether the control should move with the window when the user scrolls the window. By default, (unchecked), the control does not move with the window. Leave the **Scroll** box unchecked to create a control that stays fixed when the user scrolls the window. The **Window Formatter** places the SCROLL attribute on the control when checked.

Freeze

"Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color field or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your control declaration.

Text color only affects button text when an icon is present. However, Text color always affects the focus indicator (dotted rectangle that appears when the button has focus).

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Icon

In the **Icon** field, optionally select a standard icon, .BMP, or .GIF file. This displays a small bitmap on the button face (clipping or centering the bitmap as necessary).

To select a standard icon, choose one of the named items in the drop-down list. To select an icon file (whose extension must be .ICO), choose **Select File** from the drop-down list, then pick the file using the standard file dialog.

Immediate

Lets you create a button control which repeats the executable action continuously, for as long as the user holds the button down. Normally, buttons generate an event only after the user presses *and releases* the mouse. See also: the IMM attribute in the Language Reference manual.

Flat

(the FLAT attribute) creates a button control which appears flat until the mouse cursor passes over it. This is typically used on toolbar buttons. This feature works best if the ICON attribute names a .GIF file.

Required

Specifies that when pressed, your program automatically checks that all entry controls with the REQ attribute are neither blank nor zero. A button with this attribute is a 'required fields check' button.

Specify this type of button when a window also contains an ENTRY or TEXT control field with the REQ attribute (or else use the INCOMPLETE() function to test the ENTRY controls). When the user presses a button with the REQ attribute and an ENTRY field is blank or zero, the first required control which is blank or zero receives the focus.

Default Button

"Presses" the button when the user presses the ENTER key. A heavy border appears around the button at runtime, to signal the default button to the user. In general, place the DEFAULT attribute on a button if it represents the most likely action the user will wish to carry out. Place only one default button in a window.

Delay

The amount of time before the second EVENT:Accepted occurs in hundredths of seconds. The first EVENT:Accepted occurs when the end user clicks and holds the button. (The DELAY

Repeat

This is how often the EVENT:Accepted occurs in hundredths of seconds when the end user clicks and holds the button. (The REPEAT attribute)

STD ID

Executes a standard action when the end user presses the button. See also: STD in the Language Reference manual.

Drop ID

To specify the type of Drag operations this control will accept, type up to 16 *signatures*, separated by commas. The **Window Formatter** adds the **DROPID** attribute to the control, which indicates the control is a valid target for the drag and drop operations identified by the signatures.

Help

Cursor

The *Cursor* field (the CURSOR attribute) lets you specify an alternate shape for the cursor when the user passes the mouse over the button. The drop-down list provides standard cursor choices such as I-Beam and Crosshair. To select an external cursor file (whose extension must be .CUR), choose **Select File** from the drop-down list, then pick the file using the standard file dialog.

Help ID The **Help ID** field (the HLP attribute) takes a string constant specifying the key for accessing a specific topic in the Help document. This may be either a Help keyword or a context string.

A Help keyword is a word or phrase indexed so that the user may search for it in the *Help* **Search** dialog. When you fill in the HLP attribute for a button, if the button has focus, when the user presses F1, the help file opens to the referenced topic. If more than one topic matches a keyword, the search dialog appears.

When referencing a context string in the **Help ID** field, you must identify it with a leading tilde (~).

Message The **Message** field (the MSG attribute) lets you specify text to display in the first zone of the status bar when the control has focus.

Tip The **TIP** attribute on a control specifies the text to display in a "balloon help" box when the mouse cursor pauses over the control. Although there is no specific limit on the number of characters, the *string* should not be longer than can be displayed on the screen.

Position

Lets you set the location and size of the control.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position and size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the control's **Top Left Corner**, set an **X** (horizontal) and **Y** (vertical) coordinate in Dialog units. The coordinate is relative to the top left corner of the structure containing it, that is, the window or the report band.

To set the control's size, choose from the following options for **Width** and **Height**.

Default The Clarion runtime library determines the size of the control based on the applicable font and picture token.

Full The control is the full height or width of the window or report.

Fixed Lets you set a precise width or height in Dialog units on a window, or in thousandths of inch, points, or millimeters on a report.

Actions

Use the Actions tab to provide functionality to your button. Filling in these prompts causes the button to execute an action when the user presses the button.

When Pressed From the drop down list, choose *Call a Procedure, Run a Program*, or *No Special Action*.

Tip

Choose No Special Action if the button is from a control template.

The procedure or program you specify executes when the user pushes the button. The choices are:

Call a Procedure You must specify the **Procedure Name**, and whether the procedure will **Initiate a Thread**.

Procedure Name From the **Procedure Name** drop down list, choose an existing procedure name, or type a new procedure name. A new procedure appears as a "ToDo" item in your Application Tree.

Initiate a Thread Optionally check the **Initiate a Thread** box. If the procedure initiates a thread, specify the Thread Stack size. Clarion uses the START function to initiate a new execution thread. You can additionally specify **Parameters**, **Requested File Action**, or both.

Tip

A BUTTON on an application frame toolbar that calls an MDI child procedure <u>must</u> initiate a thread.

Thread Stack Accept the default value in the **Thread Stack** spin box unless you have extraordinary program requirements. To change the value, type in a new value or click on the spin box arrows.

Parameters In the **Parameters** field, optionally type a list of variables or data structures passed to the procedure. You are limited to the number of passed parameters supported by the START statement.

Requested File Action

From the drop down list, optionally select ***None, Insert, Change, Delete***, or ***Select***. The default selection is ***None***. The Global Request variable gets the selected value. The called procedure can then check the value of the Global Request variable and perform the requested file action.

Run a Program You must specify the **Program Name**, and optionally, any parameters.

Program Name Type the program name. The program must reside in a .DLL or .LIB defined in your application's project (.PRJ) file.

Parameters Optionally type a list of values that are passed to the program.

No Special Action Choose this option if you are providing your button's functionality with another method, such as embedded source, or an STD ID (see *Extra Tab* above).

Note:

You may combine a procedure or program call with embedded source, but not with an STD ID.

Tables Accesses the **Table Schematic Definition** dialog for this procedure.

Embeds Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for the selected control or window. As you choose options, the dialog box displays a sample of the selected font.

Check Box Control Properties

The check box provides an attractive way to display a yes/no choice for a record field--the alternative might be an entire column that repeats "one," "yes," or even ".T." for each record.

General

Text Specify a string constant to display by typing it in the **Text** box.

Use This defines the USE attribute for the control. Specify a numeric variable. The check box places a value of 1 in the numeric variable if the end user turns on the check box, zero if off.

True Value Type the value to assign when the box is checked.

False Value Type the value to assign when the box is cleared.

These values are also used to set the state of the check box (checked or cleared) when it is first displayed.

True Value and **False Value** let you easily manage legacy data with a check box, or let you use character values such as "T" and "F" or "Yes" and "No" where appropriate. For example, if your legacy field contains "True" and "False" or "Y" and "N," rather than 1 and 0, then **True Value** and **False Value** can modify the check box's default behavior to be consistent with the legacy data. If you leave both fields blank, you get the default values and behavior, that is, 1 for checked and 0 or blank for cleared.

Tip

True Value and False Value are case sensitive, so "True" is not the same as "TRUE" and "T" is not the same as "t."

Justification **Left Justification** arranges the check box (or icon) to the left of the parameter text. **Right Justification** arranges the check box (or icon) to the right of the parameter text. **Default** arranges the check box according to any applicable settings in the data dictionary.

Layout Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Mode:

Hide Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control

Skip Instructs the **Window Formatter** to omit the control from the Tab Order. When the user TABS from field to field in the dialog box, Windows will not give the control focus. This is useful for seldom-used data fields. The **Window Formatter** will place the SKIP attribute on the control. .

Scroll Specifies whether the control should move with the window when the user scrolls the window. By default, (unchecked), the control does not move with the window. Leave the **Scroll** box unchecked to create a control that stays fixed when the user scrolls the window. The **Window Formatter** places the SCROLL attribute on the control when checked.

Transparent Specify whether you wish the control background to be **Transparent**. This instructs Windows to suppress the rectangular region around the text--the background. Normally, Windows will paint this the same uniform color as the window below the control. This adds the TRN attribute.

Freeze "Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color or Background fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR to your control declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Icon

In the **Icon** field, optionally select a standard icon or icon file. This displays a small bitmap next to the check box (clipping or centering the bitmap as necessary).

To select a standard icon, choose one of the named items in the drop-down list. To select an icon file (whose extension must be .ICO), choose **Select File** from the drop-down list, then pick the file using the standard file dialog.

Drop ID

To specify the type of Drag operations this control will accept, type up to 16 *signatures*, separated by commas. The **Window Formatter** adds the **DROPID** attribute to the control, which indicates the control is a valid target for the drag and drop operations identified by the signatures.

Help

Cursor

The *Cursor* field (the CURSOR attribute) lets you specify an alternate shape for the cursor when the user passes the cursor over the control. The drop-down list provides standard cursor choices such as I-Beam and Crosshair. To select an external cursor file (whose extension must be .CUR), choose **Select File** from the drop-down list, then pick the file using the standard file dialog.

Help ID

The **Help ID** field (the HLP attribute) takes a string constant specifying the key for accessing a specific topic in the Help document. This may be either a Help keyword or a context string.

A Help keyword is a word or phrase indexed so that the user may search for it in the *Help* **Search** dialog. When you fill in the HLP attribute for a button, if the entry box has focus, when the user presses F1, the help file opens to the referenced topic. If more than one topic matches a keyword, the search dialog appears.

When referencing a context string in the **Help ID** field, you must identify it with a leading tilde (~).

Message

The **Message** field (the MSG attribute) lets you specify text to display in the first zone of the status bar when the control has focus.

Tip

The **TIP** attribute on a control specifies the text to display in a "balloon help" box when the mouse cursor pauses over the control. Although there is no specific limit on the number of characters, the *string* should not be longer than can be displayed on the screen.

Position

Lets you set the location and size of the control.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position and size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the control's **Top Left Corner**, set an **X** (horizontal) and **Y** (vertical) coordinate in Dialog units. The coordinate is relative to the top left corner of the structure containing it, that is, the window or the report band.

To set the control's size, choose from the following options for **Width** and **Height**.

Default	The Clarion runtime library determines the size of the control based on the applicable font and picture token.
Full	The control is the full height or width of the window or report.
Fixed	Lets you set a precise width or height in Dialog units on a window, or in thousandths of inch, points, or millimeters on a report.

Actions

The Check Box **Actions** tab leads to other dialogs allowing you to name variables and change their values when the end user checks or unchecks the box. Additionally, you can HIDE or UNHIDE other controls in the window.

Two group boxes with two pairs of buttons appear on the **Actions** tab. These buttons set the behavior for **When the Check Box is Checked**, and **When the Check Box is Unchecked**.

Assign Values	Opens the **Assign Values** dialog where you can assign values to variables based on the checked or unchecked state of the check box.
Hide/Unhide Controls	Opens the **Hide/Unhide Controls** dialog where you can specify window controls to hide or unhide based on the checked or unchecked state of the check box.
Tables	Accesses the **Table Schematic Definition** dialog for this procedure.
Embeds	Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for the selected control or window. As you choose options, the dialog box displays a sample of the selected font.

Assign Values Dialog

Lets you assign values to variables based on the checked or unchecked state of a check box. You may specify multiple assignments. Press the **Insert** button to add a new assignment.

Variable to Assign In the entry box, type a variable name, or press the ellipsis (...) button to choose or create a data dictionary field or a memory variable with the **Select Field** dialog.

Value to Assign In the entry box, type the value to assign to the variable. You can then add code to your program to take appropriate action based on the run time value of the variable(s).

Hide/Unhide Controls Dialog

Lets you specify window controls to hide or unhide based on the checked or unchecked state of a check box. You may specify multiple controls to hide/unhide. Press the **Insert** button to add a new hide/unhide action to the list.

Control to hide/unhide

From the drop down list, choose the control to HIDE or UNHIDE.

Hide or unhide control

From the drop down list, choose **Hide** or **Unhide**.

Combo Box Control Properties

The Combo Box combines an entry box with a list box. It is useful for when you expect string data which *usually* should be a member of the list, but which also might not be. The **Window Formatter** lets you create either a normal combo box, or a drop-down combo box.

This section only discusses *placing* the combo box. After you place it, you must format it. See the **List Box Formatter** dialog for more information on formatting and adding additional functionality to your combo boxes.

General

Picture
Specify the picture token for the control. The picture token you specify appears in the format string, for example, "@S10@." Pressing the ellipsis button lets you select the picture token from the **Edit Picture String** Dialog.

Use
This defines the USE attribute for the control. Place a variable or field equate label in the **Use** field. You may specify the variable which will receive the value that the user selects. Or, a field equate label to reference the combo box in program statements.

From
Fill the **From** field with the origin of the list data. Generally, this is the label of a QUEUE structure.

Drop
Specifies whether this should be a regular or drop-down combo box. Place a zero in the **Drop** field for a normal combo box. To create a drop-down combo box, type the number of drop-down elements you wish to be visible. You must resize the combo box after specifying the drop number.

Justification
Specify left, center, right, decimal, or default justification. Default justification matches that specified in the data dictionary, if applicable. If you use decimal justification, you set the Offset to allow display of digits to the right of the decimal point.

Offset
Specify an indentation value for the list box item text, in dialog units.

Layout
Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Mode:

Hide Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control

Skip Instructs the **Window Formatter** to omit the control from the Tab Order. When the user TABS from field to field in the dialog box, Windows will not give the control focus. This is useful for seldom-used data fields. The **Window Formatter** will place the SKIP attribute on the control. .

Scroll Specifies whether the control should move with the window when the user scrolls the window. By default, (unchecked), the control does not move with the window. Leave the **Scroll** box unchecked to create a control that stays fixed when the user scrolls the window. The **Window Formatter** places the SCROLL attribute on the control when checked.

Transparent Specify whether you wish the control background to be **Transparent**. This instructs Windows to suppress the rectangular region around the text--the background. Normally, Windows will paint this the same uniform color as the window below the control. This adds the TRN attribute.

Freeze "Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color, Background, Selected text, Selected fill, or Grid Color fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your control declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Mark Optionally enables multiple item selection. Type in the name of a QUEUE, field or array in the **Mark** field if you wish to allow the user to select more than one item from the list. The QUEUE field will flag the selected items.

VCR Optionally provide VCR controls. Check the **VCR** check box to provide VCR style controls for the list box. Optionally type the name of an entry field to the right of the check box. When the user presses the *Locator* (?) button, the focus shifts to that field. The user may type in data, then press TAB to scroll the list box to the closest matching entry.

Case Specify case attributes for the entry field portion of the combo box. The entry box can automatically convert characters from one case to another. **Uppercase** automatically converts to all caps. **Capitalize** converts to proper case. **Default** (no attribute) accepts input in the case the user types it.

Entry Mode Choose either **Insert**, **Overwrite** or **Default**. The **Entry Mode** applies only for windows with the MASK attribute set. Default accepts input according to the current system settings.

Flat Check this box to give your list box a "flat" appearance (the list box control is on the same level as the window, and not recessed). Due to a built-in attribute to the runtime library, list boxes used in Drop List and Drop Combo controls are always flat and cannot be modified in the Formatter. The entry control in the Drop Combo (and the button in both Drop controls) is not affected by this attribute.

Required - (the REQ attribute) specifies that the control may not be left blank or zero.

Read Only - (the READONLY attribute) prevents data entry in this control. Use this to declare display-only data.

Select Columns Check this box if you wish to allow the user to highlight the list component of a multi-column combo box field by field (rather than one row at a time). This provides for spreadsheet grid style movement of the highlight bar. See also: the COLUMN attribute in the Language Reference manual.

Hide Selection Specifies the selection bar appears only when the list box has focus. See also: the NOBAR attribute in the Language Reference manual.

Immediate

To generate a message event each time the end user moves or resizes the selection bar, check the **Immediate** box. This adds the IMM attribute to the COMBO. You are responsible for the code that executes upon notification of the event.

Scroll Bars

To add a horizontal scroll bar to your list box, mark the **Horizontal** check box. Scroll bars only appear when the list of items in the list is bigger than the window. To add a vertical scroll bar, check the **Vertical** check box. These options add the HSCROLL, VSCROLL, and HVSCROLL attributes to the control.

Drag ID

To specify the type of Drag operations this control will generate, type up to 16 *signatures*, separated by commas. The **DRAGID** attribute specifies the REGION control can serve as a drag-and-drop host. DRAGID works in conjunction with the **DROPID** attribute.

Drop ID

To specify the type of Drag operations this control will accept, type up to 16 *signatures*, separated by commas. The **Window Formatter** adds the **DROPID** attribute to the control, which indicates the control is a valid target for the drag and drop operations identified by the signatures.

Help

Cursor

The *Cursor* field (the CURSOR attribute) lets you specify an alternate shape for the cursor when the user passes the cursor over the control. The drop-down list provides standard cursor choices such as I-Beam and Crosshair. To select an external cursor file (whose extension must be .CUR), choose **Select File** from the drop-down list, then pick the file using the standard file dialog.

Help ID

The **Help ID** field (the HLP attribute) takes a string constant specifying the key for accessing a specific topic in the Help document. This may be either a Help keyword or a context string.

A Help keyword is a word or phrase indexed so that the user may search for it in the *Help* **Search** dialog. When you fill in the HLP attribute for a button, if the entry box has focus, when the user presses F1, the help file opens to the referenced topic. If more than one topic matches a keyword, the search dialog appears.

When referencing a context string in the **Help ID** field, you must identify it with a leading tilde (~).

Message

The **Message** field (the MSG attribute) lets you specify text to display in the first zone of the status bar when the control has focus.

Tip The **TIP** attribute on a control specifies the text to display in a "balloon help" box when the mouse cursor pauses over the control. Although there is no specific limit on the number of characters, the *string* should not be longer than can be displayed on the screen.

Position

Lets you set the location and size of the control.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position and size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the control's **Top Left Corner**, set an **X** (horizontal) and **Y** (vertical) coordinate in Dialog units. The coordinate is relative to the top left corner of the structure containing it, that is, the window or the report band.

To set the control's size, choose from the following options for **Width** and **Height**.

Default The Clarion runtime library determines the size of the control based on the applicable font and picture token.

Full The control is the full height or width of the window or report.

Fixed Lets you set a precise width or height in Dialog units on a window, or in thousandths of inch, points, or millimeters on a report.

Actions

Files Accesses the **File Schematic Definition** dialog for this procedure.

Embeds Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for the selected control or window. As you choose options, the dialog box displays a sample of the selected font.

Ellipse Control Properties

The ellipse control lets you place a circle or ellipse in your window or report. You may fill the ellipse with a color, and specify a border color. The ellipse control cannot receive focus, nor can it generate events.

General

Use

This defines the USE attribute for the control. Type a field equate label to reference the control in executable code.

Layout

Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Line Width

Set the thickness of the ellipses' border by entering a point value in the **Line Width** spin control. The default is 1 point.

Mode:

Hide

Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable

Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control.

Scroll

Specifies whether the control should move with the window when the user scrolls the window. By default, (unchecked), the control does not move with the window. Leave the **Scroll** box unchecked to create a control that stays fixed when the user scrolls the window. The **Window Formatter** places the SCROLL attribute on the control when checked.

Freeze "Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Border Color or Fill Color fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your control declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Position

Lets you set the location and size of the control.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position and size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the control's **Top Left Corner**, set an **X** (horizontal) and **Y** (vertical) coordinate in Dialog units. The coordinate is relative to the top left corner of the structure containing it, that is, the window or the report band.

To set the control's size, choose from the following options for **Width** and **Height**.

Default The Clarion runtime library determines the size of the control based on the applicable font and picture token.

Full The control is the full height or width of the window or report.

Fixed Lets you set a precise width or height in Dialog units on a window, or in thousandths of inch, points, or millimeters on a report.

Actions

Tables Accesses the **Table Schematic Definition** dialog for this procedure.

Embeds Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

Entry Box Control Properties

An entry box lets you process data input from the user. The data entry control is a specialized form of Windows edit box. It can help you automatically validate data as the user enters it in a dialog box.

General

Picture

The Picture field takes a display picture token that specifies input format. You may press the ellipsis (...) button next to the field to pick a display picture from the **Edit Picture String** dialog.

You may check the user entry against the picture at two points: as the user types the data in, or when the user closes the dialog box. Checking the data as the user types it incurs a slight performance penalty. To do so, check the **Entry Patterns** box in the **Window Properties** dialog for the window in which the entry box resides. This turns the MASK attribute on for *all* controls in the window.

If the MASK attribute is off, entry checking takes place when the user moves the focus to another control (for example, by TABBING to another field).

If the user types in data in a format different from the picture, the program will attempt to determine the format, then convert it to match the picture (if no MASK was specified). For example, if the user types 'January 1, 1995' and the picture is @D1, the program formats the input to "1/1/95. If the program cannot determine the entry format, it will *not* update the USE variable. The user will receive an audible prompt (beep), and the focus will return to the entry control, ready for additional input.

Use

This defines the USE attribute for the control. Place a variable or field equate label in the **Use** field. You may specify a variable which receives the value that the user types. Or, a field equate label that references the entry box in program statements.

Justification

Specify left, center, right, decimal, or default justification. Default justification matches that specified in the data dictionary, if applicable. If you use decimal justification, you set the Offset to allow display of digits to the right of the decimal point.

Offset

Specify an indentation value for the text, in dialog units. The indention is in the opposite direction from the justification.

Layout Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Mode:

Hide Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control.

Skip Instructs the **Window Formatter** to omit the control from the Tab Order. When the user TABS from field to field in the dialog box, Windows will not give the control focus. This is useful for seldom-used data fields. The **Window Formatter** will place the SKIP attribute on the control.

Scroll Specifies whether the control should move with the window when the user scrolls the window. By default, (unchecked), the control does not move with the window. Leave the **Scroll** box unchecked to create a control that stays fixed when the user scrolls the window. The **Window Formatter** places the SCROLL attribute on the control when checked.

Transparent Specify whether you wish the control background to be **Transparent**. This instructs Windows to suppress the rectangular region around the text--the background. Normally, Windows will paint this the same uniform color as the window below the control. This adds the TRN attribute.

Freeze "Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color, Background, Selected text, or Selected fill fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your control declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Case

Specify case attributes for the entry field. The entry box can automatically convert characters from one case to another. **Uppercase** automatically converts to all caps. **Capitalize** converts to proper case. **Default** (no attribute) accepts input in the case the user types it.

Entry Mode

Choose either **Insert**, **Overwrite** or **Default**. The **Entry Mode** applies only for windows with the MASK attribute set. Default accepts input according to the current system settings.

Options

Set the Entry flags. There are three entry flags you may toggle on or off independently.

Required - (the REQ attribute) specifies that the control may not be left blank or zero.

Immediate - (the IMM attribute) specifies immediate event generation whenever the user presses any key. See Also: “How to complete an entry field when the last character is entered” in the FAQ documentation.

Read Only - (the READONLY attribute) prevents data entry in this control. Use this to declare display-only data.

Password - (the PASSWORD attribute) specifies non-display of data entered in this control. When the user types in data, asterisks are displayed for each character entered.

Drop ID

To specify the type of Drag operations this control will accept, type up to 16 *signatures*, separated by commas. The **Window Formatter** adds the **DROPID** attribute to the control, which indicates the control is a valid target for the drag and drop operations identified by the signatures.

Help

Cursor

The *Cursor* field (the CURSOR attribute) lets you specify an alternate shape for the cursor when the user passes the cursor over the control. The drop-down list provides standard cursor choices such as I-Beam and Crosshair. To select an external cursor file (whose extension must be .CUR), choose **Select File** from the drop-down list, then pick the file using the standard file dialog.

Help ID

The **Help ID** field (the HLP attribute) takes a string constant specifying the key for accessing a specific topic in the Help document. This may be either a Help keyword or a context string.

A Help keyword is a word or phrase indexed so that the user may search for it in the *Help* **Search** dialog. When you fill in the HLP attribute for a button, if the entry box has focus, when the user presses F1, the help file opens to the referenced topic. If more than one topic matches a keyword, the search dialog appears.

When referencing a context string in the **Help ID** field, you must identify it with a leading tilde (~).

Message

The **Message** field (the MSG attribute) lets you specify text to display in the first zone of the status bar when the control has focus.

Tip

The **TIP** attribute on a control specifies the text to display in a "balloon help" box when the mouse cursor pauses over the control. Although there is no specific limit on the number of characters, the *string* should not be longer than can be displayed on the screen.

Position

Lets you set the location and size of the control.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position and size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the control's **Top Left Corner**, set an **X** (horizontal) and **Y** (vertical) coordinate in Dialog units. The coordinate is relative to the top left corner of the structure containing it, that is, the window or the report band.

To set the control's size, choose from the following options for **Width** and **Height**.

Default The Clarion runtime library determines the size of the control based on the applicable font and picture token.

Full The control is the full height or width of the window or report.

Fixed Lets you set a precise width or height in Dialog units on a window, or in thousandths of inch, points, or millimeters on a report.

Actions

The **Actions** tab prompts are all from the templates, in other words, the prompts you see here vary with the template used to create the control. Following are the standard action prompts for all entry controls.

The standard **Actions** prompts are designed to provide data validation support for your entry controls. The tab is divided into two parallel sections. The **When the Control is Selected** section provides validation when the control *receives* focus (when the user TABS onto, or mouse CLICKS the control). The **When the Control is Accepted** section provides data validation when the control *loses* focus after data have been entered in it. The control loses focus when the user TABS off the control, mouse CLICKS to a different control or window, or closes the window without canceling. The two sections are not mutually exclusive, so you can provide validation at both points.

Lookup Key Type a key label from the *lookup* file, or press the ellipsis (...) button to select a key from the **Select Key** dialog.

A lookup file is a file which contains all the valid values for the entry field, which are directly accessible through a unique.

For example, a file containing all of the customer numbers for your application could be a lookup file. The key label could be CUS:KeyCustNumber.

Tip

This lookup validation works best with a single component unique key.

Lookup Field Type the label of a component field of the lookup key, or press the ellipsis (...) button to select a field from the **Select component from key** dialog.

This is the field within the key that contains the same value being validated. Ideally, this field is the only component of a unique key.

Lookup Procedure

Type a procedure name, or choose an existing procedure from the drop down list.

This is the procedure that is called when the user enters an invalid value, and the lookup specified above *fails*. The usual purpose of this procedure is to allow the user to choose a valid value from the lookup file.

Select procedures (or Browse procedures) generated by Clarion's Wizards) are appropriate for this purpose. Alternatively, you may hand-code a procedure.

Procedure Parameters

Allows you to specify parameter names (an optional list of variables separated by commas) for your update procedure, which you can pass to it from the calling browse procedure. You must specify the functionality for the parameters in embedded source code.

Example: (LOC:HideID,GLO:AccessLevel)

Advanced

Calls the **Embedded Source** dialog. The only embed point shown is after the code generated to call the lookup procedure specified above. For more embed points, and further customization, press the **Embeds** button.

Perform lookup during non-stop select

Checking this box tells Clarion to perform the validation when the *window* is accepted, even if the *entry control* never received focus. From a practical viewpoint, checking this box prevents the user from entering blanks by virtue of having pressed the window's "**OK** button" without ever TABBING or CLICKING onto the entry field.

This option is only applicable to the **When the Control is Accepted** section.

Force Window Refresh when Accepted

Checking this box ensures that everything (including formulas and other entry fields) on the window is current and up-to-date when the user TABS off this entry control.

More Field Assignments

Optionally press the **More Field Assignments** button to specify additional value assignments from the selected lookup item's record.

Tables	Accesses the **Table Schematic Definition** dialog for this procedure.
Embeds	Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for the selected control or window. As you choose options, the dialog box displays a sample of the selected font.

Group Control Properties

A Group control places a box around two or more controls. It visually associates the controls for the user, and lets you address all the controls as one entity -- making it easy, for example, to disable all at once.

General

Text Specify a string constant by typing it in the **Text** box.

Use This defines the USE attribute for the control. Type a field equate label to reference the control in executable code, or the name of a variable

Layout Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Mode:

Hide Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control.

Skip Instructs the **Window Formatter** to omit the control from the Tab Order. When the user TABS from field to field in the dialog box, Windows will not give the control focus. This is useful for seldom-used data fields. The **Window Formatter** will place the SKIP attribute on the control.

Scroll

Specifies whether the control should move with the window when the user scrolls the window. By default, (unchecked), the control does not move with the window. Leave the **Scroll** box unchecked to create a control that stays fixed when the user scrolls the window. The **Window Formatter** places the SCROLL attribute on the control when checked.

Transparent

Specify whether you wish the control background to be **Transparent**. This instructs Windows to suppress the rectangular region around the text--the background. Normally, Windows will paint this the same uniform color as the window below the control. This adds the TRN attribute.

Freeze

"Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color or Background fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR to your control declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Bevel

Gives a three-dimensional look to the control. The box appears raised, depressed, or both. The **Window Formatter** adds the BEVEL attribute to your control declaration. Fine tune the bevel with the following prompts:

Outer

A positive value makes the control appear raised above the plane of the window. The higher the value, the further the box is raised. A negative value makes the control appear depressed below the plane of the window. The bevel effect begins at the outer border of the box.

Inner

A positive value makes the control appear raised above the plane of the window. The higher the value, the further the box is raised. A negative value makes the control appear depressed below the plane of the window. The bevel effect begins immediately inside the outer bevel.

Style An integer constant or constant expression that specifies fine control of the bevel, overriding the signs of the outer and inner parameters (PROP:BevelStyle).

Boxed Specifies whether to draw a visible box, containing the caption, around the grouped controls. When not checked, the Group box, including its caption, is invisible.

Drop ID To specify the type of Drag operations this control will accept, type up to 16 *signatures*, separated by commas. The **Window Formatter** adds the **DROPID** attribute to the control, which indicates the control is a valid target for the drag and drop operations identified by the signatures.

Help

Cursor The *Cursor* field (the CURSOR attribute) lets you specify an alternate shape for the cursor when the user passes the cursor over the control. The drop-down list provides standard cursor choices such as I-Beam and Crosshair. To select an external cursor file (whose extension must be .CUR), choose **Select File** from the drop-down list, then pick the file using the standard file dialog.

Help ID The **Help ID** field (the HLP attribute) takes a string constant specifying the key for accessing a specific topic in the Help document. This may be either a Help keyword or a context string.

A Help keyword is a word or phrase indexed so that the user may search for it in the *Help* **Search** dialog. When you fill in the HLP attribute for a button, if the entry box has focus, when the user presses F1, the help file opens to the referenced topic. If more than one topic matches a keyword, the search dialog appears.

When referencing a context string in the **Help ID** field, you must identify it with a leading tilde (~).

Message The **Message** field (the MSG attribute) lets you specify text to display in the first zone of the status bar when the control has focus.

Tip The **TIP** attribute on a control specifies the text to display in a "balloon help" box when the mouse cursor pauses over the control. Although there is no specific limit on the number of characters, the *string* should not be longer than can be displayed on the screen.

Position

Lets you set the location and size of the control.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position and size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the control's **Top Left Corner**, set an **X** (horizontal) and **Y** (vertical) coordinate in Dialog units. The coordinate is relative to the top left corner of the structure containing it, that is, the window or the report band.

To set the control's size, choose from the following options for **Width** and **Height**.

Default	The Clarion runtime library determines the size of the control based on the applicable font and picture token.
Full	The control is the full height or width of the window or report.
Fixed	Lets you set a precise width or height in Dialog units on a window, or in thousandths of inch, points, or millimeters on a report.

Actions

Tables	Accesses the **Table Schematic Definition** dialog for this procedure.
Embeds	Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for the selected control or window. As you choose options, the dialog box displays a sample of the selected font.

Image Control Properties

The Image control lets you place bitmapped and vector images in a window or report. The bitmap file formats supported are .BMP, .PCX, .GIF, .ICO, .CUR and .JPG. The vector file format supported is .WMF. Clarion can support up to 16.7-million color resolution.

You must add the PALETTE attribute to a WINDOW to support custom palettes and color depths beyond the resolution of the end user's machine at runtime. For example, to support a 16.7 million color palette, the proper attribute is PALETTE(16777215).

The Image control cannot receive focus, nor can it generate events.

General

File

Select a graphics file. Type in a file name, or press the ellipsis (**...**) button to the right of the **File** field to select a graphics file using the standard **File Open** dialog.

Use

This defines the USE attribute for the control. Type a field equate label to reference the control in executable code.

Layout

Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Mode

Hide

Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control.

Scroll Specifies whether the control should move with the window when the user scrolls the window. By default, (unchecked), the control does not move with the window. Leave the **Scroll** box unchecked to create a control that stays fixed when the user scrolls the window. The **Window Formatter** places the SCROLL attribute on the control when checked.

Freeze "Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Image Mode

Stretched Specifies the image displays stretched to fill the area of the IMAGE.

Centered Specifies the image displays at its default size and is centered in the area of the IMAGE (PROP:CENTERED)

Tiled Specifies the image displays at its default size and is tiled to fill the entire area of the IMAGE (PROP:TILED).

Extra

Scroll Bars To add a horizontal scroll bar to the control, mark the **Horizontal** check box. Scroll bars only appear when the contents of the text box are bigger than the window. To add a vertical scroll bar, check the **Vertical** check box. These options add the HSCROLL, VSCROLL, and HVSCROLL attributes to the control.

Position

Lets you set the location and size of the control.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position and size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the control's **Top Left Corner**, set an **X** (horizontal) and **Y** (vertical) coordinate in Dialog units. The coordinate is relative to the top left corner of the structure containing it, that is, the window or the report band.

To set the control's size, choose from the following options for **Width** and **Height**.

Default The Clarion runtime library determines the size of the control based on the applicable font and picture token.

Full The control is the full height or width of the window or report.

Fixed Lets you set a precise width or height in Dialog units on a window, or in thousandths of inch, points, or millimeters on a report.

Actions

Tables Accesses the **Table Schematic Definition** dialog for this procedure.

Embeds Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

Line Control Properties

The line control lets you place a straight line in your window or report. You specify a color. The line control cannot receive focus, nor can it generate events.

General

Use This defines the USE attribute for the control. Type a field equate label to reference the control in executable code.

Layout Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Line Width Set the line's thickness by entering a point value in the **Line Width** spin control. The default is 1 point.

Mode

Hide Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control.

Scroll Specifies whether the control should move with the window when the user scrolls the window. By default, (unchecked), the control does not move with the window. Leave the **Scroll** box unchecked to create a control that stays fixed when the user scrolls the window. The **Window Formatter** places the SCROLL attribute on the control when checked.

Freeze "Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Colors

Enter a valid color equate in the Line Color field, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your control declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Position

Lets you set the location and size of the control.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position and size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the control's **Top Left Corner**, set an **X** (horizontal) and **Y** (vertical) coordinate in Dialog units. The coordinate is relative to the top left corner of the structure containing it, that is, the window or the report band.

To set the control's size, choose from the following options for **Width** and **Height**.

Default The Clarion runtime library determines the size of the control based on the applicable font and picture token.

Full The control is the full height or width of the window or report.

Fixed Lets you set a precise width or height in Dialog units on a window, or in thousandths of inch, points, or millimeters on a report.

Tip

A height of zero(0) specifies a horizontal line. A width of zero(0) specifies a vertical line.

Actions

Tables Accesses the **Table Schematic Definition** dialog for this procedure.

Embeds Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

List Control Properties

The List box is useful for presenting a great number of choices for the user. It can convey a large amount of data in a small area, which has led to its use as an all purpose data control. Using Clarion, you can create list boxes which look like spreadsheet grids, perform drag-and-drop tasks, and more.

This section only discusses *placing* the list box. After you place a list box, you must format it. See the **List Box Formatter** dialog for more information on formatting and adding additional functionality to your list boxes.

General

Use

This defines the USE attribute for the control. Place a variable or field equate label in the **Use** field. You may specify the variable which will receive the value that the user selects. Or, a field equate label to reference the list box in program statements.

From

Fill the **From** field with the origin of the list data. Generally, this is the label of a QUEUE structure.

Drop

Specifies whether this should be a regular or drop-down list box. Place a zero in the **Drop** field for a normal list box. To create a drop-down list box, type the number of drop-down elements you wish to be visible.

Justification

Specify left, center, right, decimal, or default justification. Default justification matches that specified in the data dictionary, if applicable. If you use decimal justification, you set the Offset to allow display of digits to the right of the decimal point.

Offset

Specify an indentation value for the list box item text, in dialog units.

Layout

Indicates the orientation of the LIST control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Mode:

Hide Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control.

Skip Instructs the **Window Formatter** to omit the control from the Tab Order. When the user TABS from field to field in the dialog box, Windows will not give the control focus. This is useful for seldom-used data fields. The **Window Formatter** will place the SKIP attribute on the control.

Scroll Specifies whether the control should move with the window when the user scrolls the window. By default, (unchecked), the control does not move with the window. Leave the **Scroll** box unchecked to create a control that stays fixed when the user scrolls the window. The **Window Formatter** places the SCROLL attribute on the control when checked.

Transparent Specify whether you wish the control background to be **Transparent**. This instructs Windows to suppress the rectangular region around the text--the background. Normally, Windows will paint this the same uniform color as the window below the control. This adds the TRN attribute.

Freeze "Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color, Background, Selected text, Selected fill, or Grid Color fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your control declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Mark Optionally enables multiple item selection. Type in the label a QUEUE field in the **Mark** field if you wish to allow the user to select more than one item from the list. The QUEUE field will flag the selected items.

VCR Optionally provide VCR controls. Check the **VCR** check box to provide VCR style controls for the list box. Optionally type the field equate label of an entry field to the right of the check box. When the user presses the *Locator* (?) button, the focus shifts to that field. The user may type in data, then press TAB to scroll the list box to the closest matching entry.

Flat Check this box to give your list box a "flat" appearance (the list box control is on the same level as the window, and not recessed). Due to a built-in attribute to the runtime library, list boxes used in Drop List and Drop Combo controls are always flat and cannot be modified in the Formatter. The entry control in the Drop Combo (and the button in both Drop controls) is not affected by this attribute.

Select Columns Check this box if you wish to allow the user to highlight a multi-column list box field by field (rather than one row at a time). This provides for spreadsheet grid style movement of the highlight bar. See also: the COLUMN attribute in the Language Reference manual.

Hide Selection Specifies the selection bar appears only when the list box has focus. See also: the NOBAR attribute in the Language Reference manual.

Immediate To generate a message event each time the end user moves or resizes the selection bar, check the **Immediate** box. This adds the IMM attribute to the LIST. You are responsible for the code that executes upon notification of the event.

Scroll Bars To add a horizontal scroll bar to your list box, mark the **Horizontal** check box. Scroll bars only appear when the list of items in the list is bigger than the window. To add a vertical scroll bar, check the **Vertical** check box. These options add the HSCROLL, VSCROLL, and HVSCROLL attributes to the control.

Drag ID To specify the type of Drag operations this control will generate, type up to 16 *signatures*, separated by commas. The **DRAGID** attribute specifies the LIST control can serve as a drag-and-drop host. DRAGID works in conjunction with the **DROPID** attribute.

Drop ID To specify the type of Drag operations this control will accept, type up to 16 *signatures*, separated by commas. The **Window Formatter** adds the **DROPID** attribute to the control, which indicates the control is a valid target for the drag and drop operations identified by the signatures.

Help

Cursor

The *Cursor* field (the CURSOR attribute) lets you specify an alternate shape for the cursor when the user passes the cursor over the control. The drop-down list provides standard cursor choices such as I-Beam and Crosshair. To select an external cursor file (whose extension must be .CUR), choose **Select File** from the drop-down list, then pick the file using the standard file dialog.

Help ID

The **Help ID** field (the HLP attribute) takes a string constant specifying the key for accessing a specific topic in the Help document. This may be either a Help keyword or a context string.

A Help keyword is a word or phrase indexed so that the user may search for it in the *Help* **Search** dialog. When you fill in the HLP attribute for a button, if the entry box has focus, when the user presses F1, the help file opens to the referenced topic. If more than one topic matches a keyword, the search dialog appears.

When referencing a context string in the **Help ID** field, you must identify it with a leading tilde (~).

Message

The **Message** field (the MSG attribute) lets you specify text to display in the first zone of the status bar when the control has focus.

Tip

The **TIP** attribute on a control specifies the text to display in a "balloon help" box when the mouse cursor pauses over the control. Although there is no specific limit on the number of characters, the *string* should not be longer than can be displayed on the screen.

Position

Lets you set the location and size of the control.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position and size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the control's **Top Left Corner**, set an **X** (horizontal) and **Y** (vertical) coordinate in Dialog units. The coordinate is relative to the top left corner of the structure containing it, that is, the window or the report band.

To set the control's size, choose from the following options for **Width** and **Height**.

Default	The Clarion runtime library determines the size of the control based on the applicable font and picture token.
Full	The control is the full height or width of the window or report.
Fixed	Lets you set a precise width or height in Dialog units on a window, or in thousandths of inch, points, or millimeters on a report.

Actions

Tables	Accesses the **Table Schematic Definition** dialog for this procedure.
Embeds	Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for the selected control or window. As you choose options, the dialog box displays a sample of the selected font.

OLE Properties

General

Use

This defines the USE attribute for the control. Type a field equate label to reference the control in executable code.

Layout

Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Sizing Mode

Select a sizing attribute from the drop-down list. The **Window Formatter** adds the attribute to the OLE declaration.

Default Adds no sizing attribute. Zoom is the default.

Clip The OLE declaration gets the CLIP attribute. The OLE object only displays what fits into the area defined by the OLE container control's AT attribute. If the object is larger than the control, only the top left corner displays.

Stretch The OLE declaration gets the STRETCH attribute. The OLE object stretches to completely fill the area defined by the OLE container control's AT attribute. The object's aspect ratio is lost.

AutoSize The OLE declaration gets the AUTOSIZE attribute. The OLE object automatically resizes when the OLE container control's AT attribute changes at runtime.

Zoom The OLE declaration gets the ZOOM attribute. The OLE object stretches to fill the area defined by the OLE container control's AT attribute. The object's aspect ratio is maintained.

Compatible Mode Check this box to specify a compatibility mode of 1 for objects that require it (such as the Windows bitmap editor). Clear this box to let the compatibility mode default to 0.

32-bit Check this box box to tell the **Window Formatter** you want to work with 32-bit objects.

This restricts the **Object Type** list to 32-bit objects and allows the **Window Formatter** to load 32-bit Ole Servers.

Control Type

Select the object type for the OLE control declaration. Select from **Ole, Document,** or **OCX**:

Ole The **Object Type** list contains registered OLE objects. The OLE declaration gets the CREATE attribute if no **Storage File** is specified or them OPEN attribute if a **Storage File** is specified.

Object Type Select from a list of registered OLE objects, such as Excel Spreadsheets, Word documents, PowerPoint Slides, etc. to CREATE or OPEN.

Tip

When you select an OLE server, the server is automatically loaded by the Window Formatter. This can be time consuming during the window design process. Design and draw your window first, then specify your OLE control last, or specify the server at run-time using property syntax rather than at design time.

Storage File Specifies the name of an OLE Compound Storage File and the object within it to OPEN. Separate the filename and the object name with a backslash and exclamation point: *FileName\!ObjectName.*

The Application Generator supplies a default value for this field when you use the development environment to create the Compound Storage File.

Tip

RIGHT-CLICK the OLE control in the Window Formatter and choose Open from the popup menu to activate the specified OLE server in open mode. This lets you build the object independent of your application, and it automatically specifies a default filename\!objectname in the Storage File field. When a Storage File is specified, the object is OPENed rather than CREATEd.

The OLE Server can access and manipulate the object in the storage file, but only through your application. The OLE Server cannot access the storage file independently of your application. Use a Storage File when you want to limit your user's access to the document.

When the object is opened, the saved version of the OLE container properties are reloaded, so properties need not be specified for OPENed objects.

If this field is blank, the OLE object is CREATEd rather than OPENed.

Document

The **Document** entry box replaces the **Object Type** list, and the **Keep synchronized with original** check box appears. The OLE declaration gets the DOCUMENT or LINK attribute.

Document

Type the fully qualified name of the document file or press the ellipsis button (...) to select the file from the standard dialog.

A document file is a file that is associated with a specific OLE server, so the application can activate that server at runtime (e.g.. MYBUDGET.XLS is associated with Excel).

If the filename is not fully qualified, the application looks for it in the current directory. The document file should be installed on the end user's machine in the specified directory.

Keep synchronized with original (Link)

Check this box to make the OLE object a link. This generates the LINK attribute for the OLE control, which tells the server to update the original file with any changes made through your application.

Clear this box to make the OLE object an embed. This generates the DOCUMENT attribute for the OLE control, which tells the server *not* to update the original file with changes made through your application.

The default DoVerb action ({PROP:DoVerb}=0) may depend on whether the object is a link or an embed.

Storage File Specify the name of an OLE Compound Storage File and the object within it to OPEN. Separate the filename and the object name with a backslash and exclamation point: *FileName\!ObjectName.*

A Storage File may be specified for embedded documents (DOCUMENT) but not for linked documents (LINK).

The Application Generator supplies a default value for this field when you use the development environment to create the Compound Storage File.

Tip

RIGHT-CLICK the OLE control in the Window Formatter and choose Open from the popup menu to activate the specified OLE server in open mode. This lets you build the object independent of your application, and it automatically specifies a default filename\!object name in the Storage File field. When a Storage File is specified, the object is OPENed rather than CREATEd.

The OLE Server can access and manipulate the object in the storage file, but only through your application. The OLE Server cannot access the storage file independently of your application. Use a Storage File when you want to limit your user's access to the document.

When the object is opened, the saved version of the OLE container properties are reloaded, so properties need not be specified for OPENed objects.

If this field is blank, the OLE object is CREATEd rather than OPENed.

OCX

The **Object Type** list contains registered OCX objects. The OLE declaration gets the CREATE or OPEN attribute.

Object Type Select from a list of registered OCX objects to CREATE or OPEN.

Storage File Specify the name of a Compound Storage File and the object within it to OPEN. Separate the filename and the object name with a backslash and exclamation point: *FileName\!ObjectName.*

Tip

RIGHT-CLICK the OLE/OCX control in the Window Formatter and choose Custom from the popup menu to activate the Custom Properties dialog for the OCX. This records any property changes and adds them as attributes to the OLE control declaration statement.

If a property is set that cannot be represented in text form, a storage file is created to store the property. The Application Generator specifies a default filename\!object name in the Storage File field.

Tip

You can create a Storage File at runtime for the OCX by issuing a

```
?OCXControl{PROP:SaveAs}='FileName\!ObjectName'
```

while the OCX is active within your program. With this technique, you can reopen an OCX in the state the user left it!

When the object is opened, the saved version of the OLE container properties are reloaded, so properties need not be specified for OPENed objects.

If you leave this field blank, the OCX object is CREATEd rather than OPENed.

Mode:

Hide Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control.

Skip Instructs the **Window Formatter** to omit the control from the Tab Order. When the user TABS from field to field in the dialog box, Windows will not give the control focus. This is useful for seldom-used data fields. The **Window Formatter** will place the SKIP attribute on the control.

Transparent Specify whether you wish the control background to be **Transparent**. This instructs Windows to suppress the rectangular region around the text--the background. Normally, Windows will paint this the same uniform color as the window below the control. This adds the TRN attribute.

Freeze

"Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color or Background fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your control declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Help

Cursor

The **Cursor** field (the CURSOR attribute) lets you specify an alternate shape for the cursor when the user passes the cursor over the control. The drop-down list provides standard cursor choices such as I-Beam and Crosshair. To select an external cursor file (whose extension must be .CUR), choose **Select File** from the drop-down list, then pick the file using the standard file dialog.

Help ID

The **Help ID** field (the HLP attribute) takes a string constant specifying the key for accessing a specific topic in the Help document. This may be either a Help keyword or a context string.

A Help keyword is a word or phrase indexed so that the user may search for it in the *Help* **Search** dialog. When you fill in the HLP attribute for a button, if the entry box has focus, when the user presses F1, the help file opens to the referenced topic. If more than one topic matches a keyword, the search dialog appears.

When referencing a context string in the **Help ID** field, you must identify it with a leading tilde (~).

Message

The **Message** field (the MSG attribute) lets you specify text to display in the first zone of the status bar when the control has focus.

Tip

The **TIP** attribute on a control specifies the text to display in a "balloon help" box when the mouse cursor pauses over the control. Although there is no specific limit on the number of characters, the text should not be longer than can be displayed on the screen.

Position

Lets you set the location and size of the control.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position and size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the control's **Top Left Corner**, set an **X** (horizontal) and **Y** (vertical) coordinate in Dialog units. The coordinate is relative to the top left corner of the structure containing it, that is, the window or the report band.

To set the control's size, choose from the following options for **Width** and **Height**.

Default	The Clarion runtime library determines the size of the control based on the applicable font and picture token.
Full	The control is the full height or width of the window or report.

Fixed Lets you set a precise width or height in Dialog units on a window, or in thousandths of inch, points, or millimeters on a report.

Actions

Tables	Accesses the **Table Schematic Definition** dialog for this procedure.
Embeds	Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for the selected control or window. As you choose options, the dialog box displays a sample of the selected font.

Option Control Properties

The **OPTION** control declares a group of RADIO controls that offer the user a list of mutually exclusive choices. The multiple RADIO controls in the OPTION structure define the choices offered to the user.

General

Text

Specify a string constant by typing it in the **Text** box. If the control is to display a variable, type a picture token in this box.

Use

This defines the USE attribute for the control. Type a field equate label to reference the control in executable code, or the name of a variable

Layout

Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Mode:

Hide

Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable

Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control.

Skip

Instructs the **Window Formatter** to omit the control from the Tab Order. When the user TABS from field to field in the dialog box, Windows will not give the control focus. This is useful for seldom-used data fields. The **Window Formatter** will place the SKIP attribute on the control.

Scroll Specifies whether the control should move with the window when the user scrolls the window. By default, (unchecked), the control does not move with the window. Leave the **Scroll** box unchecked to create a control that stays fixed when the user scrolls the window. The **Window Formatter** places the SCROLL attribute on the control when checked.

Transparent Specify whether you wish the control background to be **Transparent**. This instructs Windows to suppress the rectangular region around the text--the background. Normally, Windows will paint this the same uniform color as the window below the control. This adds the TRN_attribute.

Freeze "Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color or Background fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your control declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Bevel Gives a three dimensional look to the control. The control appears raised, depressed, or both. The **Window Formatter** adds the BEVEL attribute to your control declaration. Fine tune the bevel with the following prompts:

Outer A positive value makes the control appear raised above the plane of the window. The higher the value, the further the box is raised. A negative value makes the control appear depressed below the plane of the window. The bevel effect begins at the outer border of the box.

Inner A positive value makes the control appear raised above the plane of the window. The higher the value, the further the box is raised. A negative value makes the control appear depressed below the plane of the window. The bevel effect begins immediately inside the outer bevel.

Style An integer constant or constant expression that specifies fine control of the bevel, overriding the signs of the outer and inner parameters (PROP:BevelStyle).

Boxed The **BOXED** attribute specifies a single-track border around the OPTION structure. The parameter text of the OPTION control appears in a gap at the top of the border box. If BOXED is omitted, the *text* parameter is not displayed on screen

Drop ID To specify the type of Drag operations this control will accept, type up to 16 *signatures*, separated by commas. The **Window Formatter** adds the **DROPID** attribute to the control, which indicates the control is a valid target for the drag and drop operations identified by the signatures.

Help

Cursor The *Cursor* field (the CURSOR attribute) lets you specify an alternate shape for the cursor when the user passes the cursor over the control. The drop-down list provides standard cursor choices such as I-Beam and Crosshair. To select an external cursor file (whose extension must be .CUR), choose **Select File** from the drop-down list, then pick the file using the standard file dialog.

Help ID The **Help ID** field (the HLP attribute) takes a string constant specifying the key for accessing a specific topic in the Help document. This may be either a Help keyword or a context string.

A Help keyword is a word or phrase indexed so that the user may search for it in the *Help* **Search** dialog. When you fill in the HLP attribute for a button, if the entry box has focus, when the user presses F1, the help file opens to the referenced topic. If more than one topic matches a keyword, the search dialog appears.

When referencing a context string in the **Help ID** field, you must identify it with a leading tilde (~).

Message The **Message** field (the MSG attribute) lets you specify text to display in the first zone of the status bar when the control has focus.

Tip The **TIP** attribute on a control specifies the text to display in a "balloon help" box when the mouse cursor pauses over the control. Although there is no specific limit on the number of characters, the *string* should not be longer than can be displayed on the screen.

Position

Lets you set the location and size of the control.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position and size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the control's **Top Left Corner**, set an **X** (horizontal) and **Y** (vertical) coordinate in Dialog units. The coordinate is relative to the top left corner of the structure containing it, that is, the window or the report band.

To set the control's size, choose from the following options for **Width** and **Height**.

Default	The Clarion runtime library determines the size of the control based on the applicable font and picture token.
Full	The control is the full height or width of the window or report.
Fixed	Lets you set a precise width or height in Dialog units on a window, or in thousandths of inch, points, or millimeters on a report.

Actions

Tables	Accesses the **Table Schematic Definition** dialog for this procedure.
Embeds	Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for the selected control or window. As you choose options, the dialog box displays a sample of the selected font.

Panel Properties

General

Use

This defines the USE attribute for the control. Type a field equate label to reference the control in executable code.

Layout

Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Mode

Hide

Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable

Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control.

Scroll

Specifies whether the control should move with the window when the user scrolls the window. By default, (unchecked), the control does not move with the window. Leave the **Scroll** box unchecked to create a control that stays fixed when the user scrolls the window. The **Window Formatter** places the SCROLL attribute on the control when checked.

Freeze

"Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Fill Color field, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds FILL attribute to your control declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Bevel Gives a three-dimensional look to the control. The control appears raised, depressed, or both. The **Window Formatter** adds the BEVEL attribute to your control declaration. Fine tune the bevel with the following prompts:

Outer A positive value makes the control appear raised above the plane of the window. The higher the value, the further the box is raised. A negative value makes the control appear depressed below the plane of the window. The bevel effect begins at the outer border of the box.

Inner A positive value makes the control appear raised above the plane of the window. The higher the value, the further the box is raised. A negative value makes the control appear depressed below the plane of the window. The bevel effect begins immediately inside the outer bevel.

Style An integer constant or constant expression that specifies fine control of the bevel, overriding the signs of the outer and inner parameters (PROP:BevelStyle).

Position

Lets you set the location and size of the control.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position and size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the control's **Top Left Corner**, set an **X** (horizontal) and **Y** (vertical) coordinate in Dialog units. The coordinate is relative to the top left corner of the structure containing it, that is, the window or the report band.

To set the control's size, choose from the following options for **Width** and **Height**.

Default The Clarion runtime library determines the size of the control based on the applicable font and picture token.

Full	The control is the full height or width of the window or report.
Fixed	Lets you set a precise width or height in Dialog units on a window, or in thousandths of inch, points, or millimeters on a report.

Actions

Tables	Accesses the **Table Schematic Definition** dialog for this procedure.
Embeds	Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

Progress Control Properties

The **PROGRESS** control declares a control that displays a progress bar. This usually displays the current percentage of completion of a batch process.

If a variable is named as the USE attribute, the progress bar is automatically updated whenever the value in that variable changes. If the USE attribute is a field equate label, you must directly update the display by assigning a value (within the range defined by the RANGE attribute) to the control's PROP:progress property

General

Use

This defines the USE attribute for the control. The field equate label references the control in program statements.

Layout

Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Mode

Hide

Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable

Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control.

Scroll

Specifies whether the control should move with the window when the user scrolls the window. By default, (unchecked), the control does not move with the window. Leave the **Scroll** box unchecked to create a control that stays fixed when the user scrolls the window. The **Window Formatter** places the SCROLL attribute on the control when checked.

Transparent

Specify whether you wish the control background to be **Transparent**. This instructs Windows to suppress the rectangular region around the text--the background. Normally, Windows will paint this the same uniform color as the window below the control. This adds the TRN attribute.

Freeze

"Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color, Background, Selected text, or Selected fill fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your control declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Range

Specifies the range of values the progress bar displays. If omitted, the default range is zero (0) to one hundred (100).

Options

Check the **Vertical** check box to allow the progress control to operate from the bottom of the control to the top. If your progress control is positioned in a horizontal (left to right) display format, you should resize the progress control accordingly.

Check the **Smooth** checkbox to allow a smooth incremental display of the progress control instead of the standard "block" format.

Drop ID

To specify the type of Drag operations this control will accept, type up to 16 *signatures*, separated by commas. The **Window Formatter** adds the **DROPID** attribute to the control, which indicates the control is a valid target for the drag and drop operations identified by the signatures.

Help

Cursor

The *Cursor* field (the CURSOR attribute) lets you specify an alternate shape for the cursor when the user passes the cursor over the control. The drop-down list provides standard cursor choices such as I-Beam and Crosshair. To select an external cursor file (whose extension must be .CUR), choose **Select File** from the drop-down list, then pick the file using the standard file dialog.

Position

Lets you set the location and size of the control.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position and size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the control's **Top Left Corner**, set an **X** (horizontal) and **Y** (vertical) coordinate in Dialog units. The coordinate is relative to the top left corner of the structure containing it, that is, the window or the report band.

To set the control's size, choose from the following options for **Width** and **Height**.

Default	The Clarion runtime library determines the size of the control based on the applicable font and picture token.
Full	The control is the full height or width of the window or report.
Fixed	Lets you set a precise width or height in Dialog units on a window, or in thousandths of inch, points, or millimeters on a report.

Actions

Tables	Accesses the **Table Schematic Definition** dialog for this procedure.
Embeds	Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for the selected control or window. As you choose options, the dialog box displays a sample of the selected font.

Prompt Control Properties

The Prompt control lets you place a specialized string object on screen which automatically provides an accelerator or mnemonic access key to the next active control following the prompt.

General

Text

Specify a string constant by typing it in the **Text** box. If the control is to display a variable, type a picture token in this box.

Use

This defines the USE attribute for the control. Type a field equate label to reference the control in executable code, or the name of a variable

Justification

Specify left, center, right, or default justification. Default justification matches that specified in the data dictionary, if applicable.

Layout

Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Mode:

Hide

Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable

Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control.

Scroll

Specifies whether the control should move with the window when the user scrolls the window. By default, (unchecked), the control does not move with the window. Leave the **Scroll** box unchecked to create a control that stays fixed when the user scrolls the window. The **Window Formatter** places the SCROLL attribute on the control when checked.

Transparent Specify whether you wish the control background to be **Transparent**. This instructs Windows to suppress the rectangular region around the text--the background. Normally, Windows will paint this the same uniform color as the window below the control. This adds the TRN attribute.

Freeze "Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color or Background fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your control declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Drop ID To specify the type of Drag operations this control will accept, type up to 16 *signatures*, separated by commas. The **Window Formatter** adds the **DROPID** attribute to the control, which indicates the control is a valid target for the drag and drop operations identified by the signatures.

Help

Cursor The *Cursor* field (the CURSOR attribute) lets you specify an alternate shape for the cursor when the user passes the mouse over the PROMPT. The drop-down list provides standard cursor choices such as I-Beam and Crosshair. To select an external cursor file (whose extension must be .CUR), choose **Select File** from the drop-down list, then pick the file using the standard file dialog.

Position

Lets you set the location and size of the control.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position and size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the control's **Top Left Corner**, set an **X** (horizontal) and **Y** (vertical) coordinate in Dialog units. The coordinate is relative to the top left corner of the structure containing it, that is, the window or the report band.

To set the control's size, choose from the following options for **Width** and **Height**.

Default	The Clarion runtime library determines the size of the control based on the applicable font and picture token.
Full	The control is the full height or width of the window or report.
Fixed	Lets you set a precise width or height in Dialog units on a window, or in thousandths of inch, points, or millimeters on a report.

Actions

Tables	Accesses the **Table Schematic Definition** dialog for this procedure.
Embeds	Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for the selected control or window. As you choose options, the dialog box displays a sample of the selected font.

Radio Button Control Properties

A Radio button, also called an option button, provides the user one of a set of mutually exclusive choices. By default, a filled-in circle represents the current selection.

A group box--an OPTION structure--must always surround the radio button choices. The **Window Formatter** automatically prompts you to create this if you try to place a radio button outside an option box. When the user selects a choice, the control fills the USE variable with the Radio text, minus the ampersand indicating the accelerator key.

When you place a radio button in a blank dialog window, it forces the creation of the OPTION STRUCTURE. After activating the Radio Button tool, or choosing **Radio Button** from the **Control** menu, CLICK in the window. An option box and one radio button appear. Select the radio button and press the Properties button, or RIGHT-CLICK and select **Properties** to open the **Radio Button Properties** dialog.

General

Text

Specify a string constant to display. Place an ampersand (&) before the single character to set the accelerator key or mnemonic access character for the radio button--this underlines the label that appears next to the radio button.

Use

This defines the USE attribute for the control. The field equate label references the radio button in program statements.

Value

When the user selects a radio button, the OPTION's USE variable receives the value that you specify here. The value you enter should match the data type of the OPTION's USE variable.

If you leave the **Value** field blank, the OPTION's USE variable receives either the string found in the **Text** field, or the button number, depending on the data type of the OPTION's USE variable.

The button number corresponds to the button's position within the option box. From the **Window Formatter** choose **Edit ▸ Order Control dialog** to see the button's tab order position within the option box.

Justification

Left Justification arranges the button (or icon) to the left of the parameter text. **Right Justification** arranges the button (or icon) to the right of the parameter text. **Default** arranges the button according to any applicable settings in the data dictionary.

Layout Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Mode

Hide Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control.

Skip Instructs the **Window Formatter** to omit the control from the Tab Order. When the user TABS from field to field in the dialog box, Windows will not give the control focus. This is useful for seldom-used data fields. The **Window Formatter** will place the SKIP attribute on the control.

Scroll Specifies whether the control should move with the window when the user scrolls the window. By default, (unchecked), the control does not move with the window. Leave the **Scroll** box unchecked to create a control that stays fixed when the user scrolls the window. The **Window Formatter** places the SCROLL attribute on the control when checked.

Transparent Specify whether you wish the control background to be **Transparent**. This instructs Windows to suppress the rectangular region around the text--the background. Normally, Windows will paint this the same uniform color as the window below the control. This adds the TRN attribute.

Freeze "Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color or Background fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your control declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Icon

In the **Icon** field (ICON), optionally select a standard icon or icon file. This displays a small bitmap on the button face (clipping or centering the bitmap as necessary).

To select a standard icon, choose one of the named items in the drop-down list. To select an icon file (whose extension must be .ICO), choose **Select File** from the drop-down list, then pick the file using the standard file dialog. At run time, the radio button appears as a "latched" 3D push button.

Drop ID

To specify the type of Drag operations this control will accept, type up to 16 *signatures*, separated by commas. The **Window Formatter** adds the **DROPID** attribute to the control, which indicates the control is a valid target for the drag and drop operations identified by the signatures.

Help

Cursor

The *Cursor* field (the CURSOR attribute) lets you specify an alternate shape for the cursor when the user passes the cursor over the control. The drop-down list provides standard cursor choices such as I-Beam and Crosshair. To select an external cursor file (whose extension must be .CUR), choose **Select File** from the drop-down list, then pick the file using the standard file dialog.

Help ID

The **Help ID** field (the HLP attribute) takes a string constant specifying the key for accessing a specific topic in the Help document. This may be either a Help keyword or a context string.

A Help keyword is a word or phrase indexed so that the user may search for it in the *Help* **Search** dialog. When you fill in the HLP attribute for a button, if the entry box has focus, when the user presses F1, the help file opens to the referenced topic. If more than one topic matches a keyword, the search dialog appears.

When referencing a context string in the **Help ID** field, you must identify it with a leading tilde (~).

Message The **Message** field (the MSG attribute) lets you specify text to display in the first zone of the status bar when the control has focus.

Tip The **TIP** attribute on a control specifies the text to display in a "balloon help" box when the mouse cursor pauses over the control. Although there is no specific limit on the number of characters, the *string* should not be longer than can be displayed on the screen.

Position

Lets you set the location and size of the control.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position and size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the control's **Top Left Corner**, set an **X** (horizontal) and **Y** (vertical) coordinate in Dialog units. The coordinate is relative to the top left corner of the structure containing it, that is, the window or the report band.

To set the control's size, choose from the following options for **Width** and **Height**.

Default The Clarion runtime library determines the size of the control based on the applicable font and picture token.

Full The control is the full height or width of the window or report.

Fixed Lets you set a precise width or height in Dialog units on a window, or in thousandths of inch, points, or millimeters on a report.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for the selected control or window. As you choose options, the dialog box displays a sample of the selected font.

Actions

Tables Accesses the **Table Schematic Definition** dialog for this procedure.

Embeds Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

Region Control Properties

General

Use

This defines the USE attribute for the control. Type a field equate label to reference the control in executable code.

Layout

Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Mode:

Hide

Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable

Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control.

Scroll

Specifies whether the control should move with the window when the user scrolls the window. By default, (unchecked), the control does not move with the window. Leave the **Scroll** box unchecked to create a control that stays fixed when the user scrolls the window. The **Window Formatter** places the SCROLL attribute on the control when checked.

Freeze

"Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Border Color or Fill Color fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your control declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Bevel

Gives a three dimensional look to the control. The control appears raised, depressed, or both. The **Window Formatter** adds the BEVEL attribute to your control declaration. Fine tune the bevel with the following prompts:

Outer A positive value makes the control appear raised above the plane of the window. The higher the value, the further the box is raised. A negative value makes the control appear depressed below the plane of the window. The bevel effect begins at the outer border of the box.

Inner A positive value makes the control appear raised above the plane of the window. The higher the value, the further the box is raised. A negative value makes the control appear depressed below the plane of the window. The bevel effect begins immediately inside the outer bevel.

Style An integer constant or constant expression that specifies fine control of the bevel, overriding the signs of the outer and inner parameters (PROP:BevelStyle).

Immediate

To generate a message event each time the mouse moves over the region, check the **Immediate** box. This adds the IMM attribute to the control. You are responsible for the code that executes upon notification of the event.

Drag ID

To specify the type of Drag operations this control will generate, type up to 16 *signatures*, separated by commas. The **DRAGID** attribute specifies the REGION control can serve as a drag-and-drop host. DRAGID works in conjunction with the **DROPID** attribute.

Drop ID

To specify the type of Drag operations this control will accept, type up to 16 *signatures*, separated by commas. The **Window Formatter** adds the **DROPID** attribute to the control, which indicates the control is a valid target for the drag and drop operations identified by the signatures.

Help

Cursor The CURSOR attribute lets you specify an alternate shape for the cursor when the user passes the cursor over the control. The drop-down list provides standard cursor choices such as I-Beam and Crosshair. To select an external cursor file (whose extension must be .CUR), choose **Select File** from the drop-down list, then pick the file using the standard file dialog.

Position

Lets you set the location and size of the control.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position and size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the control's **Top Left Corner**, set an **X** (horizontal) and **Y** (vertical) coordinate in Dialog units. The coordinate is relative to the top left corner of the structure containing it, that is, the window or the report band.

To set the control's size, choose from the following options for **Width** and **Height**.

Default The Clarion runtime library determines the size of the control based on the applicable font and picture token.

Full The control is the full height or width of the window or report.

Fixed Lets you set a precise width or height in Dialog units on a window, or in thousandths of inch, points, or millimeters on a report.

Actions

Tables Accesses the **Table Schematic Definition** dialog for this procedure.

Embeds Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

Sheet Control Properties

The **SHEET** control declares a group of TAB controls that offer the user multiple "pages" of controls for the window. The multiple TAB controls in the SHEET structure define the "pages" displayed to the user.

General

Use

This defines the USE attribute for the control. Type a field equate label to reference the control in executable code, or the name of a variable.

Tab Location

Use the drop-down list to set the location of the tabs for this SHEET. Choose from the following.

Default The tabs run across the top of the sheet.

Left The tabs run down the left side of the sheet. This adds the LEFT(width) attribute to the SHEET.

Right The tabs run down the right side of the sheet. This adds the RIGHT(width) attribute to the SHEET.

Above The tabs run across the top of the sheet. This adds the ABOVE(width) attribute to the SHEET.

Below The tabs run across the bottom of the sheet. Adds the BELOW(width) attribute to the SHEET.

Tip

To fit many tabs on a sheet, set the Justification to Above and set the Text Orientation to Up. Alternatively set the Justification to Right and set the Text Orientation to Default. Also, see Extra Tab to make scrolling tabs!

Tab Width

Specify the tab width in dialog units.

This sets the value of the width parameter for the LEFT, RIGHT, ABOVE, or BELOW attribute. By setting this width, you can make all your tabs the same size, regardless of varying text lengths per tab.

Tab width is the distance between the edges of the tab that are perpendicular to the text orientation. That is, width determines how much space appears on either end of your tab's text, not how much space appears above and below the text. This is true, regardless of text orientation.

Text Orientation

Use the drop-down list to set the orientation of the tabs and their text. A **Text Orientation** other than *Default* requires a TrueType font. Choose from the following orientations:

Default	The text reads left to right and the shape of the tab is a horizontal rectangle.
Up	The text reads from bottom to top and the shape of the tab is a vertical rectangle. This adds the UP attribute to the SHEET.
Down	The text reads from top to bottom and the shape of the tab is a vertical rectangle. This adds the DOWN attribute to the SHEET.
Inverted	The text is upside down and the shape of the tab is a horizontal rectangle. This adds the UP and the DOWN attribute to the SHEET.

Layout

Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Mode:

Hide

Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable

Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control.

Skip

Instructs the **Window Formatter** to omit the control from the Tab Order. When the user TABS from field to field in the dialog box, Windows will not give the control focus. This is useful for seldom-used data fields. The **Window Formatter** will place the SKIP attribute on the control.

Scroll

Specifies whether the control should move with the window when the user scrolls the window. By default, (unchecked), the control does not move with the window. Leave the **Scroll** box unchecked to create a control that stays fixed when the user scrolls the window. The **Window Formatter** places the SCROLL attribute on the control when checked.

Freeze

"Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color, Background, Selected text, or Selected fill fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your control declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Wizard

Hides the "tab" portion of the TAB controls.

Hiding the tabs aids in creating a wizard. A wizard is a window with a "tabless" SHEET control containing one or more TABS. You'll need to write the code to handle the "turning of the pages". See *How to Create a Wizard*. Also see the CW\EXAMPLES\APPS\WIZDEMO\WIZ.APP application.

Tip

Do not check this box until you are finished designing the window!

Spread

Resizes the tabs on the TABs to fill all the available space on the SHEET.

The resizing algorithm considers the length of the text displayed on each tab, the number of tabs, and the available space on the property sheet.

NoSheet

Check this box to erase the borders of the tab pages so that only the protruding selectable portion of the tab is visible.

This has the additional effect of making tabs located above the sheet, fall to the bottom of the sheet, and making tabs located below the sheet rise to the top of the sheet. This adds the NOSHEET attribute to the SHEET. See the *Language Reference* for more information.

Scrolling

Use the drop-down list to specify tabs scrolling behavior by choosing one of the following selections:

No Scrolling The tabs do not scroll. This is the default.

Joined Scroll Buttons
Tabs are scrollable, with adjacent scroll buttons. Adds the JOINED attribute to the SHEET. See the *Language Reference* for more information.

Spread Scroll Buttons
Tabs are scrollable, with scroll buttons at opposite ends of the sheet. Adds the HSCROLL attribute to the SHEET. See the *Language Reference* for more information.

Drop ID

To specify the type of Drag operations this control will accept, type up to 16 *signatures*, separated by commas. The **Window Formatter** adds the **DROPID** attribute to the control, which indicates the control is a valid target for the drag and drop operations identified by the signatures.

Help

Cursor

The *Cursor* field (the CURSOR attribute) lets you specify an alternate shape for the cursor when the user passes the cursor over the control. The drop-down list provides standard cursor choices such as I-Beam and Crosshair. To select an external cursor file (whose extension must be .CUR), choose **Select File** from the drop-down list, then pick the file using the standard file dialog.

Position

Lets you set the location and size of the sheet and the tabs on the sheet.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position and size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the control's **Top Left Corner**, set an **X** (horizontal) and **Y** (vertical) coordinate in Dialog units. The coordinate is relative to the top left corner of the window.

To set the control's size, choose from the following options for **Width** and **Height**.

Default	The Clarion runtime library determines the size of the control based on the applicable font and picture token.
Full	The control is the full height or width of the window.
Fixed	Lets you set a precise width or height in Dialog units.

Actions

Tables	Accesses the **Table Schematic Definition** dialog for this procedure.
Embeds	Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for the selected control or window. As you choose options, the dialog box displays a sample of the selected font.

Spin Box Control Properties

Spin Boxes are specialized entry boxes that only accept values in a predefined range. They also provide the user with "increase" and "decrease" buttons, which many people like because they can use the mouse to change the value. The user can also type a value directly into the control.

General

Picture

The Picture field takes a display picture token that specifies input format. You may press the ellipsis (...) button next to the field to pick a display picture from the **Edit Picture String** dialog.

You may check the user entry against the picture at two points: as the user types the data in, or when the user closes the dialog box. Checking the data as the user types it incurs a slight performance penalty. To do so, check the **Entry Patterns** box in the **Window Properties** dialog for the window in which the entry box resides. This turns the MASK attribute on for *all* controls in the window.

If the MASK attribute is off, entry checking takes place when the user moves the focus to another control (for example, by TABBING to another field).

If the user types in data in a format different from the picture, the program will attempt to determine the format, then convert it to match the picture (if no MASK was specified). For example, if the user types 'January 1, 1995' and the picture is @D1, the program formats the input to "1/1/95. If the program cannot determine the entry format, it will *not* update the USE variable. The user will receive an audible prompt (beep), and the focus will return to the entry control, ready for additional input.

Use

This defines the USE attribute for the control. Place a variable or field equate label in the **Use** field. You may specify a variable which receives the value that the user selects. Or, a field equate label which references the spin box in program statements.

From

The FROM attribute is optional, but is useful for values that progress in an irregular increment. You may also wish to provide the user with strings formatted as Spin Box choices when the choices are a natural progression such as the days of the week or the months of the year. Specify a QUEUE in the **From** field. This and **Range** are mutually exclusive.

Justification Specify left, center, right, decimal, or default justification. Default justification matches that specified in the data dictionary, if applicable. If you use decimal justification, you set the Offset to allow display of digits to the right of the decimal point.

Offset Specify an indentation value for the list box item text, in dialog units.

Layout Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Mode

Hide Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control.

Skip Instructs the **Window Formatter** to omit the control from the Tab Order. When the user TABS from field to field in the dialog box, Windows will not give the control focus. This is useful for seldom-used data fields. The **Window Formatter** will place the SKIP attribute on the control.

Scroll Specifies whether the control should move with the window when the user scrolls the window. By default, (unchecked), the control does not move with the window. Leave the **Scroll** box unchecked to create a control that stays fixed when the user scrolls the window. The **Window Formatter** places the SCROLL attribute on the control when checked.

Transparent Specify whether you wish the control background to be **Transparent**. This instructs Windows to suppress the rectangular region around the text--the background. Normally, Windows will paint this the same uniform color as the window below the control. This adds the TRN attribute.

Freeze "Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color, Background, Selected text, or Selected fill fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your control declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Range Specify the upper and lower **Range** limits, and the **Step** value. Place the highest value which the control should return in the **Range Upper** field. The value should match the **Picture** field. Place the lowest acceptable value in the **Lower** field. Place the Step value--the amount by which each press of the increase or decrease buttons should change the spin box value--in the **Step** field.

Case Specify case attributes for the entry field. The entry box can automatically convert characters from one case to another. **Uppercase** automatically converts to all caps. **Capitalize** converts to proper case. **Default** (no attribute) accepts input in the case the user types it.

Entry Mode Choose either **Insert**, **Overwrite** or **Default**. The **Entry Mode** applies only for windows with the MASK attribute set. Default accepts input according to the current system settings.

Options You may toggle the following options on or off independently.

Immediate - (the IMM attribute) specifies immediate event generation whenever the user presses any key.

Required - (the REQ attribute) specifies that the control may not be left blank or zero.

Read Only - (the READONLY attribute) prevents data entry in this control. Use this to declare display-only data.

Delay

The amount of time before the second EVENT:NewSelection occurs in hundredths of seconds. The first EVENT:NewSelection occurs when the end user clicks and holds the arrow button. (The DELAY attribute)

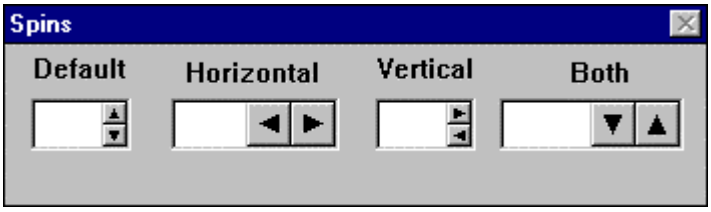
Repeat

This is how often the EVENT:NewSelection occurs in hundredths of seconds when the end user clicks and holds the arrow button. (The REPEAT attribute)

Scroll Bar

Set the appearance of the Spin control buttons by checking any combination of the two **Scroll Bar** boxes.

Checking neither box or the **Vertical** box produces smaller, vertically stacked buttons. Checking both boxes or the **Horizontal** box produces larger buttons arranged side by side.



Drop ID

To specify the type of Drag operations this control will accept, type up to 16 *signatures*, separated by commas. The **Window Formatter** adds the **DROPID** attribute to the control, which indicates the control is a valid target for the drag and drop operations identified by the signatures.

Help

Cursor

The *Cursor* field (the CURSOR attribute) lets you specify an alternate shape for the cursor when the user passes the cursor over the control. The drop-down list provides standard cursor choices such as I-Beam and Crosshair. To select an external cursor file (whose extension must be .CUR), choose **Select File** from the drop-down list, then pick the file using the standard file dialog.

Help ID

The **Help ID** field (the HLP attribute) takes a string constant specifying the key for accessing a specific topic in the Help document. This may be either a Help keyword or a context string.

A Help keyword is a word or phrase indexed so that the user may search for it in the *Help* **Search** dialog. When you fill in the HLP attribute for a button, if the entry box has focus, when the user presses F1, the help file opens to the referenced topic. If more than one topic matches a keyword, the search dialog appears.

When referencing a context string in the **Help ID** field, you must identify it with a leading tilde (~).

Message The **Message** field (the MSG attribute) lets you specify text to display in the first zone of the status bar when the control has focus.

Tip The **TIP** attribute on a control specifies the text to display in a "balloon help" box when the mouse cursor pauses over the control. Although there is no specific limit on the number of characters, the *string* should not be longer than can be displayed on the screen.

Position

Lets you set the location and size of the control.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position and size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the control's **Top Left Corner**, set an **X** (horizontal) and **Y** (vertical) coordinate in Dialog units. The coordinate is relative to the top left corner of the structure containing it, that is, the window or the report band.

To set the control's size, choose from the following options for **Width** and **Height**.

Default The Clarion runtime library determines the size of the control based on the applicable font and picture token.

Full The control is the full height or width of the window or report.

Fixed Lets you set a precise width or height in Dialog units on a window, or in thousandths of inch, points, or millimeters on a report.

Actions

Tables	Accesses the **Table Schematic Definition** dialog for this procedure.
Embeds	Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for the selected control or window. As you choose options, the dialog box displays a sample of the selected font.

String Control Properties

The String control lets you place a string constant in a window or report. It optionally lets you substitute a variable.

General

Text
For Constant Strings, specify the constant by typing it in the **Text** box.

Picture
For Variable Strings, specify the picture token for the control. Press the ellipsis button to select the picture token from the **Edit Picture String** Dialog.

Use
This defines the USE attribute for the control. Type a field equate label to reference the control in executable code, or the name of a variable. Press the ellipsis button to select or define a variable.

Justification
Specify left, center, right, decimal, or default justification. Default justification matches that specified in the data dictionary, if applicable. If you use decimal justification, you set the Offset to allow display of digits to the right of the decimal point.

Offset
Specify an indentation value for the text, in dialog units for a window, or the default measurement unit for a report. This provides the *indent* parameter for the left, center, right, or decimal, justification attribute.

Layout
Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Variable String
Optionally check the **Variable String** box. This specifies that you want to display the contents of a variable in the string control, and opens the **Select Field** dialog where you can select or define the variable. Place a picture token in the **Picture** box, such as @s24, and name the variable in the **Use** field.

Total Type

This drop down list is available only from the **Report Formatter**, and only when the **Variable String** box is checked. Choose from this drop down list to implement one of Clarion's built in report totaling functions. Choose from ***Sum, Average, Maximum, Minimum, Count,*** and ***Page No***.

To create page totals or group totals, use the **Reset On** drop down list and the **Tallies** list on the Extra tab in conjunction with your **Total Type** selection. Totals are calculated or tallied each time a structure in the **Tallies** list prints. Totals are reset each time the **Reset On** structure prints.

Using

The label of a numeric variable to receive the intermediate values calculated for the SUM, AVE, MAX,MIN, CNT, or PAGENO. This allows you to create totals on other totals. The value in the variable is internally updated by the print engine, so it is only useful for use within the REPORT structure. This is only available when a **Total Type** is specified.

Reset On

This drop down list is available only from the **Report Formatter**, and only when the **Variable String** box is checked. To create page totals or group totals, use the **Reset On** drop down list in conjunction with your **Total Type** selection. Reset your totals at the beginning of each **Page**, or at the beginning of any BREAK group within the report.

Mode

Hide

Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable

Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control.

Scroll

Valid on on a WINDOW. Specifies whether the control should move with the window when the user scrolls the window. By default, (unchecked), the control does not move with the window. Leave the **Scroll** box unchecked to create a control that stays fixed when the user scrolls the window. The **Window Formatter** places the SCROLL attribute on the control when checked.

Skip

Valid only in a REPORT. Check this box to specify not to print the control if the content is blank, and to move all following controls in the band upward to "fill in" the blank (PROP:SKIP).

Transparent Specify whether you wish the control background to be **Transparent**. This instructs Windows to suppress the rectangular region around the text--the background. Normally, Windows will paint this the same uniform color as the window below the control. This adds the TRN attribute.

Freeze "Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color or Background fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your control declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Tallies The total is calculated each time a highlighted structure in the **Tallies** list prints. This list is available only from the **Report Formatter** and only when the **Variable String** box is checked. Use the **Tallies** list in conjunction with the **Total Type** and **Reset On** lists on the General tab to generate a custom total field. The **Report Formatter** adds the TALLY attribute to the string.

Angle Specify the angle of the text. An angle other than zero requires a TrueType font. Adds the ANGLE attribute to the STRING. With this, you can rotate the text from its normal horizontal position through a full 360 degrees.

Tip

Make the control taller than usual to accommodate the slant of the text.

Extend Valid in REPORT only. Enter a valid string of attributes that are assigned to a designated REPORT control for a given document type. For more information, see EXTEND in the Language Reference Manual.

Drop ID To specify the type of Drag operations this control will accept, type up to 16 *signatures*, separated by commas. The **Window Formatter** adds the **DROPID** attribute to the control, which indicates the control is a valid target for the drag and drop operations identified by the signatures.

Help

Cursor The *Cursor* field (the CURSOR attribute) lets you specify an alternate shape for the cursor when the user passes the cursor over the control. The drop-down list provides standard cursor choices such as I-Beam and Crosshair. To select an external cursor file (whose extension must be .CUR), choose **Select File** from the drop-down list, then pick the file using the standard file dialog.

Position

Lets you set the location and size of the control.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position and size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the control's **Top Left Corner**, set an **X** (horizontal) and **Y** (vertical) coordinate in Dialog units. The coordinate is relative to the top left corner of the structure containing it, that is, the window or the report band.

To set the control's size, choose from the following options for **Width** and **Height**.

Default The Clarion runtime library determines the size of the control based on the applicable font and picture token.

Full The control is the full height or width of the window or report.

Fixed Lets you set a precise width or height in Dialog units on a window, or in thousandths of inch, points, or millimeters on a report.

Actions

Tables Accesses the **Table Schematic Definition** dialog for this procedure.

Embeds Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

Font

Calls the **Select Font** dialog which lets you change the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for the selected control or window. As you choose options, the dialog box displays a sample of the selected font.

Tab Control Properties

The **TAB** structure declares a group of controls that constitute one of the multiple "pages" of controls contained within a SHEET structure. The multiple TAB controls in the SHEET structure define the "pages" displayed to the user. The SHEET structure's USE attribute receives the *text* of the TAB control selected by the user.

General

Text

Specify a string constant by typing it in the **Text** box. If the control is to display a variable, type a picture token in this box.

Use

This defines the USE attribute for the control. Type a field equate label to reference the control in executable code, or the name of a variable.

Layout

Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Hide

Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable

Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control.

Freeze

"Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Color

Enter a valid color equate in the Text Color, Background, Selected text, or Selected fill fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your control declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Required

Specifies that when selected, your program automatically checks that all entry controls with the REQ attribute are neither blank nor zero.

Specify this type of tab when a window also contains an ENTRY or TEXT control field with the REQ attribute (or else use the INCOMPLETE() function to test the ENTRY controls). When the user clicks on a tab with the REQ attribute and an ENTRY field is blank or zero, the first required control which is blank or zero receives the focus. See also: How to Create a Multi-Page Form in the FAQ PDF.

Drop ID

To specify the type of Drag operations this control will accept, type up to 16 *signatures*, separated by commas. The **Window Formatter** adds the **DROPID** attribute to the control, which indicates the control is a valid target for the drag and drop operations identified by the signatures.

Help

Help ID

The **Help ID** field (the HLP attribute) takes a string constant specifying the key for accessing a specific topic in the Help document. This may be either a Help keyword or a context string.

A Help keyword is a word or phrase indexed so that the user may search for it in the *Help* **Search** dialog. When you fill in the HLP attribute for a button, if the entry box has focus, when the user presses F1, the help file opens to the referenced topic. If more than one topic matches a keyword, the search dialog appears.

When referencing a context string in the **Help ID** field, you must identify it with a leading tilde (~).

Message

The **Message** field (the MSG attribute) lets you specify text to display in the first zone of the status bar when the control has focus.

Tip

The **TIP** attribute on a control specifies the text to display in a "balloon help" box when the mouse cursor pauses over the control. Although there is no specific limit on the number of characters, the *string* should not be longer than can be displayed on the screen.

Actions

Tables	Accesses the **Table Schematic Definition** dialog for this procedure.
Embeds	Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for the selected control or window. As you choose options, the dialog box displays a sample of the selected font.

Text Box Control Properties

The Text control provides a multi-line data entry field. This control is especially suitable for holding a long string.

General

Use

This defines the USE attribute for the control. Place a variable or field equate label in the **Use** field. You may specify a variable which receives the value that the user types. When using multi-line controls, be sure the variable is large enough to hold the amount of data you expect your users to enter in the control. Or, type a field equate label which references the entry box in program statements.

Justification

Specify left, center, right, or default justification. Default justification matches that specified in the data dictionary, if applicable.

Layout

Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Mode:

Hide

Makes the control invisible at the time Windows would initially display it. Windows actually creates the control--it just doesn't display it on screen. The **Window Formatter** places the HIDE attribute on the control. Use the UNHIDE statement to display the control.

Disable

Disables or 'grays-out' the control when your program initially displays it. The **Window Formatter** places the DISABLE attribute on the control. Use the ENABLE statement to allow the user access to the control.

Skip Instructs the **Window Formatter** to omit the control from the Tab Order. When the user TABS from field to field in the dialog box, Windows will not give the control focus. This is useful for seldom-used data fields. The **Window Formatter** will place the SKIP attribute on the control.

Scroll Specifies whether the control should move with the window when the user scrolls the window. By default, (unchecked), the control does not move with the window. Leave the **Scroll** box unchecked to create a control that stays fixed when the user scrolls the window. The **Window Formatter** places the SCROLL attribute on the control when checked.

Transparent Specify whether you wish the control background to be **Transparent**. This instructs Windows to suppress the rectangular region around the text--the background. Normally, Windows will paint this the same uniform color as the window below the control. This adds the TRN attribute.

Freeze "Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

RTF Check this box to allow the TEXT control to read and write its contents in **Rich Text Format**. A full set of supported methods is provided to allow you to add additional features and support. See RTF.

Value Mode This drop list is enabled if you have checked the **RTF** option. Select *Field* if the actual rich text format is stored in the selected field. Select *File* if the field name is storing the file name from which the rich text format is to be read from and written to.

Color

Enter a valid color equate in the Text Color, Background, Selected text, or Selected fill fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your control declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

Case Specify case attributes for the entry field. The entry box can automatically convert characters from one case to another. **Uppercase** automatically converts to all caps. **Default** (no attribute) accepts input in the case the user types it.

Options

You may toggle the following options on or off independently.

Required - (the REQ attribute) specifies that the control may not be left blank or zero.

Read Only - (the READONLY attribute) prevents data entry in this control. Use this to declare display-only data.

Single - (the SINGLE attribute) specifies the control is only for single line data entry. This is specifically to allow use of TEXT controls instead of ENTRY controls for data entry in languages that write from right to left (such as Hebrew or Arabic).

Resize - (the RESIZE attribute) for reports only. Allows the control to increase its height (and the parent report band's height) at runtime to accommodate varying amounts of text from its USE variable.

Extend

Valid in REPORT only. Enter a valid string of attributes that are assigned to a designated REPORT control for a given document type. For more information, see EXTEND in the Language Reference Manual.

Boxed

The **BOXED** attribute specifies a single-track border around the TEXT structure. This attribute only works if the TEXT control is transparent or the parent window does not have the GRAY attribute applied.

Scroll Bars

To add a horizontal scroll bar to the control, mark the **Horizontal** check box. Scroll bars only appear when the contents of the text box are bigger than the window. To add a vertical scroll bar, check the **Vertical** check box. These options add the HSCROLL, VSCROLL, and HVSCROLL attributes to the control.

Drop ID

To specify the type of Drag operations this control will accept, type up to 16 *signatures*, separated by commas. The **Window Formatter** adds the **DROPID** attribute to the control, which indicates the control is a valid target for the drag and drop operations identified by the signatures.

Help

Cursor

The *Cursor* field (the CURSOR attribute) lets you specify an alternate shape for the cursor when the user passes the cursor over the control. The drop-down list provides standard cursor choices such as I-Beam and Crosshair. To select an external cursor file (whose extension must be .CUR), choose **Select File** from the drop-down list, then pick the file using the standard file dialog.

Help ID The **Help ID** field (the HLP attribute) takes a string constant specifying the key for accessing a specific topic in the Help document. This may be either a Help keyword or a context string.

A Help keyword is a word or phrase indexed so that the user may search for it in the *Help* **Search** dialog. When you fill in the HLP attribute for a button, if the entry box has focus, when the user presses F1, the help file opens to the referenced topic. If more than one topic matches a keyword, the search dialog appears.

When referencing a context string in the **Help ID** field, you must identify it with a leading tilde (~).

Message The **Message** field (the MSG attribute) lets you specify text to display in the first zone of the status bar when the control has focus.

Tip The **TIP** attribute on a control specifies the text to display in a "balloon help" box when the mouse cursor pauses over the control. Although there is no specific limit on the number of characters, the *string* should not be longer than can be displayed on the screen.

Position

Lets you set the location and size of the control.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the position and size visually by dragging with the mouse in the **Window Formatter**.

To set the location of the control's **Top Left Corner**, set an **X** (horizontal) and **Y** (vertical) coordinate in Dialog units. The coordinate is relative to the top left corner of the structure containing it, that is, the window or the report band.

To set the control's size, choose from the following options for **Width** and **Height**.

Default The Clarion runtime library determines the size of the control based on the applicable font and picture token.

Full The control is the full height or width of the window or report.

Fixed Lets you set a precise width or height in Dialog units on a window, or in thousandths of inch, points, or millimeters on a report.

Actions

Tables	Accesses the **Table Schematic Definition** dialog for this procedure.
Embeds	Accesses the **Embedded Source** dialog for points surrounding the event handling for this control only.

Font

Calls the **Select Font** dialog which lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for the selected control or window. As you choose options, the dialog box displays a sample of the selected font.

Toolbar Properties

Lets you edit settings for your TOOLBAR.

General

Freeze

"Freezes" the control so that subsequent data dictionary changes are not applied. You can override the #Freeze attribute for all controls or for individual controls. See **Application Options**.

Layout

Indicates the orientation of the control.

Left to Right maintains the original layout specified in the Window Formatter.

Right to Left essentially "flips" the controls' display as a mirror image of the layout specified in the Window Formatter. Default field navigation moves from right to left.

The setting in the Application Frame will cascade its setting to all child windows and controls that have the **default** setting active.

Color

Enter a valid color equate in the Text Color, Background, Selected text, or Selected fill fields, or press the ellipsis (...) button to select a color from the **Color** dialog. The **Window Formatter** adds the COLOR attribute to your control declaration.

See ..\LIBSRC\EQUATES.CLW for a list of valid color equates. See *Windows Design Issues* in the *User's Guide* for a discussion on using color to enhance your application.

Extra

No Merge

Specifies *not* to merge an MDI tool bar into an application frame tool bar at run time.

Help

Cursor The *Cursor* field (the CURSOR attribute) lets you specify an alternate shape for the cursor when the user passes the cursor over the control. The drop-down list provides standard cursor choices such as I-Beam and Crosshair. To select an external cursor file (whose extension must be .CUR), choose **Select File** from the drop-down list, then pick the file using the standard file dialog.

Position

Lets you set the size of the toolbar. The toolbar is always located at the upper left corner of the window, that is, any X and Y coordinates are ignored.

The **Position** tab lets you specify the AT attribute. Filling in the attribute manually is optional--you may set the height visually by dragging with the mouse in the **Window Formatter**.

To set the toolbar's size, choose from the following options for **Width** and **Height**.

Default The Clarion runtime library determines the size of the toolbar, about twice the height of the window's title bar and the full width of the window.

Fixed Lets you set a precise width or height in Dialog units.

If the toolbar's width is less than the width of the window, the **Window Formatter** displays a full width toolbar to remind you that the "unused" space to the right of the toolbar may be used when merging toolbars.

Actions

Tables Accesses the **Table Schematic Definition** dialog for this procedure.

Embeds Accesses the **Embedded Source** dialog for this procedure.

Font

Calls the **Select Font** dialog that lets you select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for the selected control or window. As you choose options, the dialog box displays a sample of the selected font.

Index: