Template User's 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

This document is designed to help both new and experienced users become familiar with some of the new templates introduced in Clarion 6, and also revisit some of the existing templates that have been enhanced and improved in this version.

This new Template Guide organizes each template by a specific topic. Templates that are multi or special purpose are also noted.

All of the information contained in this guide is also reflected in the online help. Wherever there is a conflict in documentation, always refer to the online help as the most recent source.

ADO Template Support

ADO Requirements

The use of Clarion with ADO requires that you have installed the Microsoft Data Access Components (MDAC) interface, which is a free download from the Microsoft web site. You must have Version 2.62 or later installed.

The general flow of using the new ADO interface in Clarion:

1. Import your ADO tables using the Dictionary Editor.
2. Add the ADO Global Extension to your application
3. Add your various ADO procedures as necessary.

ADO Code Templates

ADO Browse Refresh Code Template

The ADO Browse Refresh Code Template allows you to refresh an ADO Browse contents at nearly any source embed point. This is useful when a data value is updated, and its value directly affects the browse box contents.

The code template has no prompts, and simply calls the ADO object's Refresh method.

ADO Browse to XML Code Template

The ADO Browse to XML Code Template allows you to export the contents of an active browse box to an external XML file. The appropriate XML tags are automatically added to the file.

Press the ellipsis to select a variable that will hold the XML external filename. The variable must hold the full path and filename. If the path is omitted, the current program directory is used.

ADOCommandExecute code template

The ADOCommandExecute code template is used to call the ADO Command Object's **Execute** method.

What is actually executed is based on your settings specified in the ADO Command Object extension template.

There are no prompts provided for this template.

ADORecordsetGetBOF code template

This code template will retrieve the value of the BOF (Beginning of File) property of the active Recordset.

Press the ellipsis button to select a valid **SHORT** variable that will hold the state of the BOF property.

ADORecordsetGetCacheSize code template

This code template will retrieve the maximum number of rows that the ADO layer cache can contain.

Press the ellipsis button to select a valid **LONG** variable that will hold the CacheSize rows value.

ADORecordsetGetCursorLocation code template

This code template is used to retrieve the value of the Cursor Location property of the associated Recordset object.

Press the ellipsis button to select a valid **LONG** variable that will hold the Cursor Location property value.

ADORecordsetGetEOF code template

This code template will retrieve the value of the EOF (End of File) property of the active Recordset.

Press the ellipsis button to select a valid **SHORT** variable that will hold the state of the EOF property.

ADORecordsetGetMaxRecords code template

This code template will retrieve a number representing the maximum number of rows set for the Recordset specified.

Press the ellipsis button to select a valid **LONG** variable that will hold the maximum rows value.

ADORecordsetGetPageCount code template

This code template will use the **GetPageCount** method to retrieve the number of pages of the Recordset object's contents.

Press the ellipsis button to select a valid **LONG** variable that will hold the PageCount value.

ADORecordsetGetPageSize code template

This code template will use the **GetPageSize** method to retrieve the size (number of rows) of a Recordset object's page contents.

Press the ellipsis button to select a valid **LONG** variable that will hold the PageSize value.

ADORecordsetOpen code template

This code template will call the **Open** method of the associated Recordset object with parameters specified in the template extension.

Recordset Open Method tab

Specify the **Cursor** and **Lock** types provided in the drop list. These settings should be found in your native ADO data source documentation. You may also specify additional **Execute Options** if necessary.

Recordset Query Parameter

This section allows you to "complete" or modify the default SQL statement, provided that the associated recordset has a command type of *adCmdText*. (See the Command Type tab control in the ADO RecordSet Object Extension Template)

The purpose is to give you an opportunity to modify the default SQL statement. For example, if the recordset was set to "SELECT * FROM Customer" in order to generate the proper buffer contents, you may need to filter the contents in another procedure (e.g., load a recordset in a procedure for lookup purposes after some user input).

ADORecordsetPutCacheSize code template

This code template will set the number of rows that should exist in the internal cache of the ADO layer.

Press the ellipsis button to select a valid **LONG** variable that will hold the number of rows to write to the ADO layer.

ADORecordsetPutCursorLocation code template

This code template will set the selected Cursor location value for the cursor location property of the associated Recordset object.

Note:

This code template should be inserted *before* the open method of the recordset.

ADORecordsetPutMaxRecords code template

This code template will set the maximum number of rows that the Recordset object should retrieve.

Press the ellipsis button to select a valid **LONG** variable that will hold maximum the number of rows to write to the ADO layer.

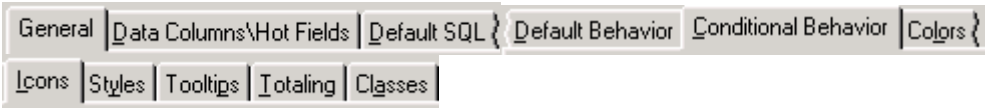
ADORecordsetPutPageSize code template

This code template will use the **PutPageSize** method to write the size (number of rows) of a Recordset object's page contents.

Press the ellipsis button to select a valid **LONG** variable that will hold the PageSize result value.

ADO Control Templates

ADO Browse Box Control Template



The ADO Browse Box Control template provides the extensive features of the standard Browse Box control, and managed through an ADO connection layer.

In addition, there are new features that extend the functionality of the standard Browse Box, including:

- Ability to change column position at runtime, by dragging the appropriate column to its new position.
- Ability to click on a column header and sort by that column at runtime. You can also SHIFT click on an additional header in order to add another column to the sort order sequence. SHIFT click again to change the order from ascending to descending.
- Ability to HIDE/UNHIDE columns by pressing CTRL+RIGHTCLICK in the data area of the ADO Browse Box at runtime.
- More powerful template-based filtering options.

The following prompts are available:

General

Loading Method

Select **Page** or **Table** from the drop list provided. **Page** tells the Browse Box to return a page of records to the client when requested. Table returns all records of the primary table used in the Browse Box, unless limited below.

Limit the result set rows number

Check this box to limit the number of rows returned by the client's request.

Maximum rows

Specify a maximum number of rows to return. This prompt is only available if the **Limit the result set rows number** check box is checked.

Generate initial call to Refresh

Check this box to force the browse to immediately refresh all data elements when the browse is first loaded. This will allow any secondary elements to be updated properly.

Connection Group:

Connection Object

Choose a Connection object name from the drop list provided. This list should contain the connection object you created in the ADO Global support template.

Use a New Connection

If you do not wish to use any of the Global Connections that are available, check this box to create a new connection specifically for this browse box.

New Connection

Press this button to call the Connection Builder. You also have the option to create (derive) a new Connection object to use instead of the global object; check the **Generate a COMIniter object** check box to enable this feature.

DB Interface Group:

DB Interface object

The Database Interface object name is generated by the ADO template. If you need to override the default object name, enter a new name here.

CRecordset Attributes

Press the **Crecordset Attributes** button to access special properties that affects the ADO recordset. Each attribute is described in detail in the ADO Reference provided by Microsoft.

Cursor Location

The **CursorLocation** property on a **Connection** object or **Recordset** object indicates the location of the cursor engine.

Cursor Type

The **CursorType** property on a **Recordset** object indicates the type of the cursor engine.

LockType

The **LockType** property is used to set the type of locks placed on records that the provider should use when opening the **Recordset**.

Command Type

The **CommandType** property on a **Recordset** object Indicates the type of a **Command** object.

Execute Options

The **Execute** method on a **Connection** object executes the statement specified in the **CommandText** property

Table Mapper – BASETABLENAME is not supported

Check this box if your ADO data source does not support BASETABLENAME capability.

In ADO, when you have created a SELECT statement through your template design that contains the same column name, but coming from a different table, the fields' collection will have field objects with the same name.

For example,

```
SELECT Customer.SysID, order.SysId etc…
```

The Fields collection above will have 2 Field objects with the name *SysID*. In order to make sure that the values from those 2 fields will go into the right application variable, we need to know which table the field belongs to. There is a property in the Field properties collection called ***BASETABLENAME*** that provide this value.

Data Columns/Hot Fields

The Data Columns and Hot Fields tab control provide prompts that allow control of the data elements that will appear in the Browse Box, in the scrollable (Data Columns) and non-scrollable (Hot Fields) areas.

The following prompts are presented:

Automatically manage repeated columns name

Check this box to allow the ADO template to handle duplicate column names automatically. If you are writing custom SQL statements, and wish to name your own repeated columns, turn off this check box.

Data Columns

The Data Columns list box displays data elements that have been populated via the List Box Formatter. Press the **Properties** button to access the following prompts:

Query Field

Identifies a column as a field that can be queried (searched).

Column is a

Identifies the data element as one that is read directly from the ADO data source (**Table Column**), or one that is a variable defined within the application, like a computed or conditional field (**Expression**).

Use AS

Check this box if the column name is long and verbose, and you need to rename the column to a more descriptive and compacted name. This is useful if you are constructing long or complex SQL statements. This is an option for Table Columns, but required for all Expressions.

Unique Field ID/AS

Enter a unique Field ID to use with the **Use AS** option for Table Columns, or the AS option for an Expression..

Expression

Enter a valid expression to use for the defined Data Column

This column cannot be hidden

Check this box to restrict this Data Column from being hidden as set in the Default Behavior tab.

This column cannot be sorted

Check this box to restrict this Data Column from being a sort field as defined in the Default Behavior tab.

This Column is the Default sort column

Check this box to indicate that this column is to be used as the default sort. This will generate a call to the cBrowse method *ApplySort* with the corresponding column number in order to perform the sort and update of the list box header accordingly.

Use the **This Column is the Default sort column** check box to set the column to use as the default locator.

Hot Fields

The Hot Fields list box displays data elements that have been populated *outside* of the List Box Formatter, or data elements that may not be populated, but need to be referenced or updated in the procedure source. Hot fields are normally data elements that are related to the contents of the list box (i.e., Address information, text based descriptions, etc.). Press the **Properties** button to access the following prompts:

Hot Field

Press the ellipsis button to select a field to use as the hot field.

Column is a

Identifies the data element as one that is read directly from the ADO data source (**Table Column**), or one that is a variable defined within the application, like a computed or conditional field (**Expression**).

Use AS

Check this box if the column name is long and verbose, and you need to rename the column to a more descriptive and compacted name. This is useful if you are constructing long or complex SQL statements. This is an option for Table Columns, but required for all Expressions.

Unique Field ID/AS

Enter a unique Field ID to use with the **Use AS** option for Table Columns, or the AS option for an Expression.

Expression

Enter a valid expression to use for the defined Data Column

Expression Data Type

Select a valid data type from the drop list that the ADO layer will translate.

Default SQL

Because the SQL to access the data is vitally important, the ADO templates allow you to view the generated SELECT statement and customize it if necessary. This also provides a convenient way to customize the browse ordering, should you wish.

Regenerate SQL
Press this button to reset the original template-constructed SQL statements. This is useful should you need to start from scratch again before customizing your statements. Only enabled when the next prompt is active (checked).

Override SELECT SQL
Check this box to bypass the auto generation of the SELECT statement by the templates. Your custom statement will be substituted in its place. This box also enables the **Regenerate SQL** button, should you wish to reset the statement back to its original value.

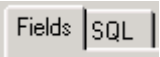
Default SQL Select
This text box provides the base SELECT statement. Note that if your browse contain fields from more than a single table, it will automatically provide for a JOIN.

Important Note: If you've populated a field in the Data Columns tab for which the ADO templates can't resolve the proper syntax (for example, if you've populated a data variable in the list box control for your browse) and the resulting page doesn't display properly, examine the SQL statement here. Should you find a statement with a blank for the field name (look for an extra comma in the order that the suspect field appears in the data columns list, as in Select fieldname,fieldname,,fieldname...), you may edit the statement here or delete the suspect column from the list. Such a problem should be very rare.

Unique Key

As stated on the tab control, ADO/SQL requires a unique key to identify a record. Should you have more than one unique key defined in your table, press the ellipsis button to select an alternate key to use.

Default Behavior



The default behavior of the ADO Browse Box has a much different interface than the standard browse box. You will also notice that the filter capabilities are expanded in the ADO template. The following prompts are available:

Fields Tab:

The Fields tab control provides control of your browse box filtering and sorting features.

Range or Filter Columns

Press the update buttons to add (Insert), modify (Properties), or remove (Delete) a range or filter column. These are the data elements used to limit the records displayed in your Browse Box. The Range or Filter dialog provides the following prompts:

Column
Select a column name from the drop list provided to use as criteria for the browse filter or range. The columns displayed are those columns you have defined in the Data Columns/Hot Fields dialog.

Use Static Value?
Check this box to use the static value of the column selected. This is the value of the column when the browse is first initialized

Negate the Range or Filter(NOT)
Check this box to apply the NOT clause to the selected column. This will have the effect of "reversing" the filter expression (Example: *MyCol = MyValue* will be generated as *WHERE NOT (MyCol = MyValue)*.

Range Limit Type
Specifies the type of range limit to apply. Choose one of the following from the drop-down list.

Single Value
Lets you limit the filter criteria to a single value. Specify the variable containing that value in the **Range Limit Value** box which appears.

Note:

In the case where you are using static values, you are responsible to enter the values with the necessary single quotes. For example, if you are using the **IN** filter criteria, and need to check several parameters enter *'Test', 'Test2','TestN'* in the **Single Value** line. For static values, you need to enter single quotes where needed.

Range of Values
Lets you specify upper and lower limits. Specify the variables containing the limits in the **Low Limit Value** and **High Limit Value** boxes.

Less Than
Lets you limit the records read to all records less than a single value. Specify the variable containing that value in the **Range Limit Value** box which appears.

Great Than
Lets you limit the records read to all records greater than a single value. Specify the variable containing that value in the **Range Limit Value** box which appears.

IN
Lets you limit the records read to all records that match the contents of a single value.

Begins with
Lets you limit the records read to all records that begin with the contents of a single value.

Ends with
Lets you limit the records read to all records that end with the contents of a single value.

Contains
Lets you limit the records read to all records that begin with the contents of a single value.

Sort Columns

Press the update buttons to add (Insert), modify (Properties), or remove (Delete) the browse sort column(s). These are the data elements used to sort the records displayed in your Browse Box. Clicking on the column header activate the column sort.

Add Primary Key Fields

Press this button to automatically include all fields defined in the primary key as sort columns.

Add a Key Field
Press this button to select any key from the primary table, and apply it to the sort columns.

The **Sort Columns** dialog also provides the following prompts:

Column
Select a column name from the drop list provided to use as criteria for the browse filter or range. The columns displayed are those columns you have defined in the Data Columns/Hot Fields dialog.

Direction
Choose ASC from the drop list to designate an ascending sort, or DESC to specify a descending sort.

SQL Tab Prompts

The SQL tab control displays SQL statements that are generated, based on the settings displayed on the Fields tab.

Essentially, the settings of the Range or Filter are used to generate the SQL WHERE clause, and the Sort Columns setting are used to generate the ORDER BY clause.

Regenerate WHERE SQL
Press this button to regenerate the template SQL WHERE statement that is constructed based on your default settings.

Override WHERE SQL
Check this box to override the default template generated WHERE clause.

Enter a WHERE clause to filter this list
The text in this box provides the Where clause, which is concatenated to the Select statement.

Regenerate ORDER BY SQL
Press this button to regenerate the template SQL ORDER BY statement that is constructed based on your default settings.

Override ORDER BY SQL
Check this box to override the default template generated ORDER BY clause.

Enter the ORDER BY clause
The text in this box provides the Order clause, which is concatenated to the other parts of the statement.

Test SQL Query
Press this button to test the SQL statement. If a statement clause is not overridden, the statement will be regenerated based on the template's current settings.

Create SQL Query

Press this button to create the SQL Query statement. If a statement clause is not overridden, the statement will be regenerated based on the template's current settings.

Conditional Behavior

The Conditional Behavior tab duplicates the prompts and resultant functions found on the Default Behavior tab, with the addition of a **Condition** prompt:

Condition
Enter any valid Clarion expression.

Colors

The Colors dialog of the ADO Browse template presents the exact functionality as the standard browse box.

Icons

The Icons dialog of the ADO Browse template presents the exact functionality as the standard browse box.

Styles

The Styles dialog of the ADO Browse template presents the exact functionality as the standard browse box.

Tooltips

The Tooltips dialog of the ADO Browse template presents the exact functionality as the standard browse box.

Totaling

The Totaling dialog of the ADO Browse template presents the exact functionality as the standard browse box.

Classes

Use the Classes tab to override the global settings for the Class. See Classes Tab.

ADO Browse Box Select Button

The ADO BrowseSelectButton template provides Select button to choose a record from An ADO browse (list) box.

The generated source code gets the currently selected record from the list (makes the selected record the current one in the browsed file's record buffer), and closes down the procedure. For the end user, pressing the Select button is equivalent to double-clicking an item in the list.

The BrowseSelectButton template provides the following prompts:

Hide the Select button when not applicable

Check this box to hide the Select button when the procedure is not called for selection purposes (GlobalRequest <> SelectRecord).

Allow Select via Popup

Check this box if you would like to include the Select option in the Browse Box popup menu.

ADO Browse Process Button

The ADO BrowseProcessButton template provides a button to call an ADO Process or Report Procedure. This control acts on the records in a browse box. When pressed, the button retrieves the recordset and invokes the respective database action for that record.

The ADO BrowseProcessButton template provides the following prompts:

Process Procedure
Type a procedure name or select a procedure name from the drop-down list. If you type a new procedure name, the Application Generator adds the new procedure to the Application Tree. The procedure that you call will expect an ADO Process Control template to exist.

Procedure Parameters
Allows you to specify parameter names (an optional list of variables separated by commas) for your process 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)

ADO Browse Update Button control template

The ADO BrowseUpdateButtons template provides three buttons for managing ADO I/O for the ADO BrowseBox: Insert, Change, and Delete. These three button controls act on the records in a browse box. When pressed, the button retrieves the selected record and invokes the respective database action for that record.

The ADO BrowseUpdateButtons template provides the following prompts:

Update Procedure
Type a procedure name or select a procedure name from the drop-down list. If you type a new procedure name, the Application Generator adds the new procedure to the Application Tree.

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)

Set UNIQUETABLE

This checkbox will control the generation of code to set the value of UNIQUETABLE in the recordset for updating. If checkbox is set, this will enable the **UNIQUEVALUE Value** prompt.

UNIQUETABLE Value

Press the ellipsis button to select the UNIQUETABLE (default value is the Primary file).

Generate a Resync statement

Check this box to force the ADO templates to generate a RESYNC statement. Some back ends (e.g., Oracle) do not need this statement specified. The statement generated will correspond to the one generated for the ADO browse box with an additional WHERE clause in which the Columns that compose the primary key are included.

Example:
```
SELECT * from Table where SysID = ?
```

Call Resync after update

Check this box if the ADO recordset method RESYNC is called after an update. If set, the prompts **AffectRecords Parameter** and **Resync Values** parameter are enabled.

AffectRecords Parameter

This prompt defines the value that must be set for the **AffectRecords** parameter of the RESYNC command. Default value is *adAffectAll*, but some back ends may only work with *adAffectCurrent* (e.g., Pervasive).

Resync Values parameter

This prompt defines the value that should be set for the **ResyncValues** parameter of the RESYNC command. Default value is *adResyncAllValues*.

ADO Errors Object List Control Template

The ADO Error Object List control template displays a list of any ADO errors that are encountered during the login and connection attempt. The list box contains two elements:

Error Number – an error code returned from the ADO Connection layer

Error Description – a detailed description of the type of error returned.

This control needs to be populated on a procedure that is named by the *ADO Error Procedure* template prompt found in the ADO Login Extension or ADO Global Extension.

ADOLoginControls Control Template

The ADOLoginControls Control Template is used with the ADO Login Procedure to provide default login controls on the window. This includes:

UserID

Password

Server

Blank Password check box option

Login and Cancel Buttons

There are no template prompts associated with this control template.

ADO Process Pause Button

This control template adds a button to allow pausing and restarting of an ADO Process or Report procedure.

Pause Text
The text to display on the button when the process is running. This text alerts the user that the process can be paused by pressing the button. The default is *Pause.*

Restart Text
The text to display on the button when the process is paused and multiple starts are allowed. This text alerts the user that pressing the button can restart the process. The default is *Restart.*

Start Paused
The state of the control when the procedure starts. If check, the process is paused until the user presses the button.

Start Text
The text to display on the button when the procedure opens. This text alerts the user that the process can be strated by pressing the button. The default is *Go.*

Allow Multiple Starts
Check this button to allow the process to start more than once.

When Pressed
The standard set of prompts for buttons. Normally, when using a Control template, these prompts are not used.

ADO Save Button Control Template

The ADO Save Button Control Template is used to update a series of data elements through an ADO connection. There are several option that are unique and different from the standard Save Button used in ISAM and SQL database updates.

In the Save Button properties (found on the button's Actions tab control) press the **ADO/SQL Behavior** button to access the ADO/SQL Behavior dialog. The following prompts are available.

General Tab

Connection Group:

Connection Object

Choose a Connection object name from the drop list provided. This list should contain the connection object you created in the ADO Global support template.

Use a New Connection

If you do not wish to use any of the Global Connections that are available, check this box to create a new connection specifically for this browse box.

New Connection

Press this button to call the Connection Builder. You also have the option to create (derive) a new Connection object to use instead of the global object; check the **Generate a COMIniter object** check box to enable this feature.

Command Object
The ADO connection layer contains a default command object that is responsible for handling the appropriate data update behavior. If you need to select an alternative object name, enter the value here.

Columns Tab

The Columns tab control prompts are used to specify additional columns (fields) that need to be updated by the ADO Form, but are not populated on the window.

By default, all data elements populated on the window that are part of the ADO file will be automatically updated as needed.

Get records affected?

Check this box if you want to retrieve the number of records affected after an update command is issued by the ADO command object. This number will be the same number you would see if you issued an SQL update command and the backend responded with "*x* records were affected by the command.

In most cases, the records affected variable will return a 1 if the update to the ADO table was successful. A zero (0) will be returned if the update to the ADO table was not successful.

Records affected var

Select a variable name to store and process the Records Affected count.

Additional Columns to add in SQL statement

In addition to the fields (columns) populated on the ADO Form procedure window, you can add additional columns to update here. For example, you may need to add data to selected columns based on data entered on the form, but need to do it in source and hide it from the user.

SQL Tab

Override Generated SELECT
Check this box to bypass the auto generation of the SELECT statement by the templates. Your custom statement will be substituted in its place. This box also enables the **Regenerate SQL** button, should you wish to reset the statement back to its original value.

Regenerate SELECT
Press this button to reset the original template-constructed SQL statements. This is useful should you need to start from scratch again before customizing your statements. Only enabled when the next prompt is active (checked).

Classes Tab

The Classes Tab contains information regarding the names of classes used by the ADO form templates. For more information, see Classes Tab.

Help on the other Save button prompts are identical to the standard Save Button control template, and can be referenced there.

ADO Extensions

SoftVelocity ADO Support Template - Global Extension

This release of Clarion introduces template support for ADO (Active Data Objects). Your browses and forms can operate with ADO like they would with conventional databases. The big difference is that ADO uses OLE DB/ODBC as the connection means, instead of the native drivers. The ADO templates are designed to provide a similar design interface, as you are already familiar with using conventional database interfaces.

The ADO Global Extension is required for any application that will use ADO. In addition, you can set up Global Connections to ADO that can be used by any ADO-based procedure.

Connections

The ADO Support Global Extension is used to add ADO connections to use throughout your application. Press the appropriate update buttons to add (**Insert**), modify (**Properties**), or remove (**Delete**) an ADO connection

In the Connections dialog window the following prompts are available:

Connection Object

Enter a name (string) that you wish to use to reference your connection. You can override the default name that is provided by the template.

Connection Class

The Connection Class references the default library used to connect to the ADO service. If you wish to use an alternate third-party class, or a class that you have written, select it from the drop list provided. Otherwise, accept the default class (Cconnection)

Cursor Location

A cursor is a database element that controls record navigation, update ability of data, and the visibility of changes made to the database by other users.

Cursors are set and based on the type of data you anticipate working with, and where the processing will take place.

Don't set
This tells the ADO connection to use the default cursor method of the particular data source. Unless your site has specific requirements that need to override this, leave this setting as is.

AdUseClient
This value indicates that a client-side cursor supplied by a local cursor library is to be used.

AdUseServer
This value indicates that the data provider or driver-supplied cursor is used.

AdUseClientbatch
For backward compatibility, this value indicates that a client-side cursor supplied by a local cursor library is to be used.

Note that the cursor settings drop list choices are all prefaced with "ad;" these are simply the ADO constant names. The descriptions of each cursor type are provided as "basic" information about the cursor. *Please refer to the documentation provided by your OLE DB Provider vendor and/or DBMS for specific information about your site.*

Connection String – Build From Source

Every ADO application requires a connection string. Select from one of 5 sources to build your connection string from:

From a Login Procedure

Choose this connection source option if you want to create a login procedure for your users to connect to the ADO source. You must use the **ADOLogin procedure** template provided in the ADO template set. The ADOLogon procedure will identify the connection string source (dictionary, UDL file, etc.)

Enter the name of the login procedure to use here.

From Dictionary

Your data dictionary stores ADO connection string information for each file that you imported using the ADO file driver. Press the ellipsis button to choose a **Dictionary Table**. The connection string will be displayed in the text box provided.

You also have the option to save the string read from the Dictionary into a Universal Data Link (.udl) file. Press the **Save Connection String in UDL file** button to write the string contents to an external UDL file.

Note:

The first two lines of any universal data link (.udl) file must have exactly this content:

```
[oledb]
; Everything after this line is an OLE DB initstring
```

After these required lines, the remainder of the .udl file consists of a connection string written in the same format used when opening a database from programming code.

UDL file read at runtime

Press the ellipsis button to load an existing UDL file to use as the connection source at runtime.

UDL file picked and read in the template

Select this option if you wish to load an existing UDL file through the current template interface, modify the connection string, and save the modified connection string to a new external UDL file.

Press the ellipsis button to load an existing UDL filename's connection string into the connection builder text window. After your changes are made, you can save the changes to a new UDL file name by pressing the **Save Connection String in UDL file** button.

It is not required to load an existing UDL file. You can create a custom connection string from scratch, or paste it from an alternative source.

Connection Builder

Select this source option to manually build your connection string using the Conncection Builder. There is an option to save this string to an external UDL file.

Override DCT Connection String
The Connection dialog window automatically reads the first ADO connection string it finds in the active data dictionary. Check this box to override this default connection string and build another in the **Connection String** text box. Unchecking this box will force a refresh of the default dictionary connection string, and overwrite your changes.

Connection String

Press the **Connection Builder** button to create an ADO connection to a designated data source. The contents (output) of the Connection Builder will be displayed in the Connection String text box. You also have the option of entering a custom (manual) connection string here.

Call Connection Builder

The Connection Builder button calls the Data Link Properties dialog. The specifics of this dialog vary according to which OLE DB provider you choose. You may choose items and fill in four tabs within this dialog.

Note that after calling the Connection Builder, the connection string appears at the bottom of the Connection Properties dialog. The following is a sample of an MS SQL connection for the Northwind sample database using MS SQL Server's OLE DB provider:

```
Provider=SQLOLEDB.1;Password=xxxxxxx;Persist Security
Info=True;User ID1=UserID;Initial Catalog=northwind;Data
Source=servername
```

Save Connection String in UDL file

Press the **Save Connection String in UDL file** button to write the current connection string to an external UDL file. You can later use this file as your connection source in multiple applications.

Classes

The Classes tab contains an extensive list of default classes used with the ADO templates. Unless you specifically create a new class definition, or, obtain one from a third-party vendor, you should accept the default class settings.

Add ADO File Extension

The ADO File Extension allows you to include additional ADO file declarations that you have declared within your data dictionary.

Including this extension in any procedure creates a "To Do" in the Table Schematic Definition named "Other ADO File". This interface allows you to include and reference any element of the ADO table without forcing the templates to generate a connection.

Add Other Files too?

Check this box to allow related ADO file definitions to be included with the ADO file that you have added. These files will also be included without forcing the templates to generate a connection

ADO Command Object Extension Template

The ADO Command Object extension template is used to give you access to the ADO Command Object and its method from any procedure. The extension will generate the necessary class declaration and code to create an ADO command object

You must have the Global ADO Support template included in your application.

The following prompts are provided:

Command Object Name and **Command Class**
These prompts simply hold the default object and class names defined by the ADO Command Object. If you have overwritten these values in the source files, you can enter different names here.

Connection Settings

In most cases you should use the default connection provided by the Global support template. However, there may be times when you need to use the ADO Command Object to send a command to a different server or database.

Check this box to use an existing connection object, and select a valid name from the drop list.

Command Timeout

When set to zero (0), the ADO Command Object will use default internal settings, based on the type of command executed. On certain occasions, you may need to override this command setting for certain stored procedures.

Recordset Information

As in the Command Object **Name** and **Class** options, the default RecordSet object and class names are defaulted here. If you have overwritten these values in the source files, you can enter different names here.

You may also specify if the Recordset returned will be stored in a GROUP or QUEUE. This option simply tailors the proper prototype parameters to use in the ADO Command Object.

Command Type

In the **Command Type** tab, depending of the type of command that is selected, there is a button used to get a list of available stored procedures (*adCmdStoredProc*). The command type selected can also execute a query to see if the query is valid (*adCmdText*).

The types of commands here are standard ADO commands that are found in any standard ADO reference guide. Consult your auxiliary documentation for more information.

Note:

In order to actually execute this command, you will have to include the ADOCommandExecute code template into an appropriate embed point.

ADO Login Extension Template

The ADO Login Extension template is used with the ADO Login procedure to establish the method and source of connecting to an ADO source.

The following template prompts are provided:

Login Procedure builds the connection string from:

There are essentially two sources of Connection String storage.

1. Select **UserID, Password, and Server prompt at runtime** if the user will be required to login each time the application is loaded (the ADO Login procedure is always the first procedure called by default).

Login details

Number of retries allowed

Enter a number here to set the number of user login attempts.

Display a message if retries exceeded

Check this box to pop up a message box if the user has exceeded the number of login attempts.

Message Box caption
Enter text that will be displayed in the message window's title bar.

Message Text
Enter the message text to display to the user when login attempts have been exceeded.

Return from procedure after Message call
Check this box to automatically return after closing the message window

Connection string data

After the User is validated, the connection string is generated from one of the following methods:

From Dictionary
Select a table name from the drop list provided. If the table is a valid ADO file type, the connection string will automatically be read and displayed in the text box the follows.

From UDL file
If the user login is successful, the ADO connection string will be read from an external UDL (Universal Data Link) file name that you specify here. Manually enter a valid UDL file, or use the ellipsis to select one from the popup file dialog.

Connection Builder
Select this option, and press the **Connection Builder** button to invoke the data link properties interface and build your ADO connection string. The connection string will be stored by the Login Procedure.

Your entry
Select this option If you wish to manually enter a connection string, or paste it from an alternative source, and a store it locally in the Login Procedure.

2. If you select **UDL File** as the connection string source, a user login is not required. The application attempts to read from the UDL file name you specify.

   Check the **Data Link Dialog can be displayed to the user** check box to allow a connection dialog to be displayed when attempting to connect to the data source. If left unchecked, the data connection process is silent (no indication of connection attempt is shown to the user). You should only leave this check box unchecked if you are sure that your ADO connection is from a reliable source.

   Call an Error procedure if Connection failed
   Check this box if you would like to call a special procedure that will display a custom message and possible reasons for the connection failure. In the **ADO Error Procedure** drop combo, enter the name of the procedure that will be used, or select from an existing procedure in the drop list.

ADO RecordSet Object Extension Template

The ADO template extension ADORecordsetObject will generate the necessary class and code to create an ADO recordset object.

The following prompts are provided:

Recordset Object Name and **Class**

These prompts simply hold the default object and class names defined by the ADO Recordset Object. If you have overwritten these values in the source files, you can enter the different names here.

Generate a TableMapper object

Check this box to instruct the templates to generate a TableMapper object. This object is used to map the ADO column names accessed to the buffer elements defined in your application.

Mapping is normally automatic, and you should not need to use this option, unless you need to manually map the Recordset.

Generate buffer as a

You must specify if the Recordset returned will be stored in a GROUP or QUEUE. This option simply tailors the proper prototype parameters to use in the ADO Command Object.

Check the **Generate loop to fill the queue after Open call** check box if you need to return multiple record sets into the QUEUE.

Connection Settings

In most cases you should use the default connection provided by the Global support template. However, there may be times when you need the ADORecordset Object to reference a different server or database.

Check this box to use an existing connection object, and select a valid name from the drop list.

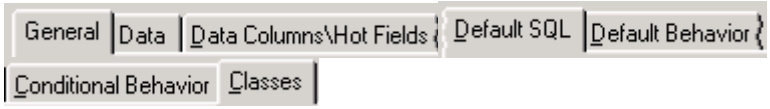
In the **Command Type** tab, depending of the type of command that is selected, there is a button used to get a list of available stored procedures (*adCmdStoredProc*). The command type selected can also execute a query to see if the query is valid (*adCmdText*).

The types of commands here are standard ADO commands that are found in any standard ADO reference guide. Consult your auxiliary documentation for more information.

Note:

In order to actually execute this command, you will have to include the ADORecordsetOpen code template in an appropriate embed point near the beginning of the procedure.

ADO Process Extension



The ADO Process Extension Template is the core template for all Process and Report procedures that use ADO as a connection layer.

The following prompts are available:

General Tab

Process Prompts

Processed Text
Enter a string of up to 100 characters that will be displayed on the process progress window. If left blank, the default is "Processed"

Timer Value

Enter a numeric value to specify the time slicing interval between the time that rows are being processed, and the update frequency of the progress window. Values are in hundredths of a second. If left blank, the default value is 50 (half second).

Rows to fetch per timer

Enter a number to designate how many rows can be processed in each timer interval. This number should be decreased if the complexity of the process is increased. The default value is 10.

Estimated Record Count
If the number of rows to be processed will be consistent, enter an estimated row count here to allow the progress bar to display the processing more accurately. If left at zero, the process engine will make a best guess based on filtering and other elements.

Close Window

Check this box to automatically close the progress window when the ADO process has completed.

Enable Report

Check this box to call a report after the process is completed. This check box will in effect enable the procedure's Report button and the subsequent Report Formatter.

Print Preview

Check this box to call a Print Preview window prior to printing the report.

Detail Filters Prompts

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. For example, to force the template to not print a detail, enter FALSE.

Optionally, check the **Exclude unfiltered** box to restrict any other detail band which does not have a filter exprerssion of its own.

Preview Options Prompts

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.

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.

Data

The Data tab control contains prompts that focus on the ADO connection information:

Process Recordset as a parameter
Check this box to disable the ADO connection process (methods) for this procedure. This implies that the information (Recordset) that needs to be processed will be passed to the procedure.

Connection Group:

Connection Object

Choose a Connection object name from the drop list provided. This list should contain the connection object you created in the ADO Global support template.

Use a New Connection

If you do not wish to use any of the Global Connections that are available, check this box to create a new connection specifically for this browse box.

New Connection

Press this button to call the Connection Builder. You also have the option to create (derive) a new Connection object to use instead of the global object; check the **Generate a COMIniter object** check box to enable this feature.

Command Object
The ADO connection layer contains a default command object that is responsible for handling the appropriate data update behavior. If you need to select an alternative object name, enter the value here.

Data Columns/Hot Fields

The Data Columns and Hot Fields tab control provide prompts that allow control of the data elements that will appear in the Process or Report procedure.

The following prompts are presented:

Hot Fields

The Hot Fields list box displays data elements that may not be populated in the report structure, but need to be referenced or updated in the procedure source. Hot fields are normally data elements that are related to the contents of the primary data elements (i.e., Address information, text based descriptions, etc.). Press the **Properties** button to access the following prompts:

Hot Field

Press the ellipsis button to select a field to use as the hot field.

Column is a

Identifies the data element as one that is read directly from the ADO data source (**Table Column**), or one that is a variable defined within the application, like a computed or conditional field (**Expression**).

Use AS

Check this box if the column name is long and verbose, and you need to rename the column to a more descriptive and compacted name. This is useful if you are constructing long or complex SQL statements. This is an option for Table Columns, but required for all Expressions.

Unique Field ID/AS

Enter a unique Field ID to use with the **Use AS** option for Table Columns, or the AS option for an Expression..

Expression

Enter a valid expression to use for the defined Data Column

Expression Data Type
Select a valid data type from the drop list that the ADO layer will translate.

Data Columns

The Data Columns list box displays data elements that have been populated in the Report Formatter. Press the **Properties** button to access the following prompts:

Query Field
Identifies a column as a field that can be queried (searched).

Column is a
Identifies the data element as one that is read directly from the ADO data source (**Table Column**), or one that is a variable defined within the application, like a computed or conditional field (**Expression**).

Use AS
Check this box if the column name is long and verbose, and you need to rename the column to a more descriptive and compacted name. This is useful if you are constructing long or complex SQL statements. This is an option for Table Columns, but required for all Expressions.

Unique Field ID/AS
Enter a unique Field ID to use with the **Use AS** option for Table Columns, or the AS option for an Expression..

Expression
Enter a valid expression to use for the defined Data Column

Default SQL

Because the SQL to access the data is vitally important, the ADO templates allow you to view the generated SELECT statement and customize it if necessary. This also provides a convenient way to customize the process or report ordering, should you wish.

Regenerate SQL
Press this button to reset the original template-constructed SQL statements. This is useful should you need to start from scratch again before customizing your statements. Only enabled when the next prompt is active (checked).

Override SELECT SQL
Check this box to bypass the auto generation of the SELECT statement by the templates. Your custom statement will be substituted in its place. This box also enables the **Regenerate SQL** button, should you wish to reset the statement back to its original value.

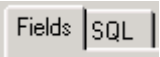
Default SQL Select
This text box provides the base SELECT statement. Note that if your process or report contains fields from more than a single table, it will automatically provide for a JOIN.

Important Note: If you've populated a field in the Data Columns tab for which the ADO templates can't resolve the proper syntax and the resulting page doesn't print properly, examine the SQL statement here. Should you find a statement with a blank for the field name (look for an extra comma in the order that the suspect field appears in the data columns list, as in Select fieldname,fieldname,,fieldname...), you may edit the statement here or delete the suspect column from the list.

Unique Key

As stated on the tab control, ADO/SQL requires a unique key to identify a record. Should you have more than one unique key defined in your table, press the ellipsis button to select an alternate key to use.

Default Behavior



The default behavior of the ADO Process control has a much different interface than the standard report or process. You will also notice that the filter capabilities are expanded in the ADO template. The following prompts are available:

Fields Tab:

The Fields tab control provides control of your process or report's filtering and sorting features.

Range or Filter Columns

Press the update buttons to add (Insert), modify (Properties), or remove (Delete) a range or filter column. These are the data elements used to limit the records processes in the Process control. The Range or Filter dialog provides the following prompts:

Column
Select a column name from the drop list provided to use as criteria for the Process/report's filter or range. The columns displayed are those columns you have defined in the Data Columns/Hot Fields dialog.

Use Static Value?
Check this box to use the static value of the column selected. This is the value of the column when the process is first initialized

Negate the Range or Filter(NOT)
Check this box to apply the NOT clause to the selected column. This will allow records to be selected only if the column value evaluates to a blank or zero.

Range Limit Type
Specifies the type of range limit to apply. Choose one of the following from the drop-down list.

Single Value
Lets you limit the filter criteria 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 variables containing the limits in the **Low Limit Value** and **High Limit Value** boxes.

Less Than
Lets you limit the records read to all records less than a single value. Specify the variable containing that value in the **Range Limit Value** box which appears.

Great Than
Lets you limit the records read to all records greater than a single value. Specify the variable containing that value in the **Range Limit Value** box which appears.

IN
Lets you limit the records read to all records that match the contents of a single value.

Begins with
Lets you limit the records read to all records that begin with the contents of a single value.

Ends with
Lets you limit the records read to all records that end with the contents of a single value.

Contains
Lets you limit the records read to all records that begin with the contents of a single value.

Sort Columns

Press the update buttons to add (Insert), modify (Properties), or remove (Delete) the process controls default sort elements. These are the data elements used to alternately sort the records processed by the control. The SQL tab control generates the ORDER BY clause.

Add Primary Key Fields
Press this button to automatically include all fields defined in the primary key as sort columns.

Add a Key Field
Press this button to select any key from the primary table, and apply it to the sort columns.

The **Sort Columns** dialog also provides the following prompts:

Column
Select a column name from the drop list provided to use as criteria for the browse filter or range. The columns displayed are those columns you have defined in the Data Columns/Hot Fields dialog.

Direction
Choose ASC from the drop list to designate an ascending sort, or DESC to specify a descending sort.

SQL Tab Prompts

The SQL tab control displays SQL statements that are generated, based on the settings displayed on the Fields tab.

Essentially, the settings of the Range or Filter are used to generate the SQL WHERE clause, and the Sort Columns setting are used to generate the ORDER BY clause.

Regenerate WHERE SQL
Press this button to regenerate the template SQL WHERE statement that is constructed based on your default settings.

Override WHERE SQL
Check this box to override the default template generated WHERE clause.

Enter a WHERE clause to filter this list
The text in this box provides the Where clause, which is concatenated to the Select statement.

Regenerate ORDER BY SQL
Press this button to regenerate the template SQL ORDER BY statement that is constructed based on your default settings.

Override ORDER BY SQL
Check this box to override the default template generated ORDER BY clause.

Enter the ORDER BY clause
The text in this box provides the Order clause, which is concatenated to the other parts of the statement.

Test SQL Query
Press this button to test the SQL statement. If a statement clause is not overridden, the statement will be regenerated based on the template's current settings.

Create SQL Query
Press this button to create the SQL Query statement. If a statement clause is not overridden, the statement will be regenerated based on the template's current settings.

Conditional Behavior

The Conditional Behavior tab duplicates the prompts and resultant functions found on the Default Behavior tab, with the addition of a **Condition** prompt:

Condition
Enter any valid Clarion expression.

Classes

Use the Classes tab to override the global settings for the Class. See Classes Tab.

ADO Procedure Templates

ADO Browse Procedure

The ADO Browse Procedure is used to display a list of records in a standard list box, using an ADO connection layer to retrieve the data elements (or recordset). The default procedure is a combination of the ADO Browse Box control, ADO BrowseUpdate control, and a standard Close Button control. The ADO Browse procedure is supported for both ABC and Clarion template chains.

See: SoftVelocity ADO Browse Box, ADO Browse Update Buttons

ADO Form

The ADO Form procedure includes the template elements necessary to allow updates of an active ADO connection.

See: ADO Save Button Control

ADO Login Procedure Template

The ADO Login Procedure Template provides a quick and easy way to allow ADO connections using a variety of different techniques.

Two templates are associated with this procedure. For more information refer to the following topics:

See:

ADO Login Controls

ADO Login Extension Template

ADO Process Procedure

The ADO Process Procedure contains the template elements necessary to call a process that affects an existing ADO connection. An optional Pause ADO Process Button can also be added to the existing defaults.

See:

ADO Process Control Extension

ADO Pause Process Button Control

ADO Report Procedure

The ADO Report Procedure contains the template elements necessary to call a process that returns, and prints, a recordset of an existing ADO connection. The ADO Report Procedure is nearly identical to the ADO Process Procedure template, with the addition of a default report structure, and preview/print logic. An optional Pause ADO Process Button can also be added to the existing defaults.

See:

ADO Process Control Extension

ADO Pause Process Button Control

Advanced Report Generation

Clarion 6 introduces a powerful new set of templates and underlying classes that provide redirection of your printed reports to several new document formats.

To implement advanced report generation, include the appropriate extension (linked below) to your **Global** section of your application. Next, you can modify, enable, and disable each target document output on the procedure level of each report. The report to document procedure template has options and controls unique to each specific format. Each report procedure's **Report Properties** include a Report Targets tab option to designate default output targets and different runtime options.

Advanced report generation supports multiple document output capability for each target report. For example, you can have one report that only redirects output to a PDF (Portable Document File) file format, another report that redirects output to HTML and XML only, based on the user's selection. You can even configure another report to let the user choose between all formats: PDF, HTML, XML, plain text (TXT), and the usual report output directly to printer!

Specifications on the web:

HTML	http://www.w3.org/MarkUp/
PDF	http://partners.adobe.com/asn/developer/acrosdk/docs.html#filefmtspecs
XML	http://www.w3.org/XML/

Report to HTML Support Extension

The Report to HTML Global Extension template is part of a suite of Advanced Report Generation tools available. This template include the underlying code support to allow you to direct any report's output defined in your application to an HTML document file. By including this template in the Global section of your application, each report procedure is automatically populated with support for this output type.

The following prompts are provided:

Disable this output

Check this box to make this template inactive in all report procedures. Use this feature when troubleshooting any potential conflicts between other template add-ons and third party products.

Html Global String Settings

Press this button to access default text strings that will be used in your HTML output. More help can be found by referring to the topics below.

See Also: Advanced Report Generation Overview

Report Targets

HTML Global String Settings

Report to HTML Procedure Support

HTML Global String Settings

HTML String settings are used to set defaults for HTML text standards. Global Settings apply to all reports, but you can always override them at the individual procedure level.

Your HTML document can output multiple pages based on the PAGE size and details of the report. Navigation is built into the templates to move from page to page.

The following text prompts are provided:

For each prompt, you can enter string text, or use a variable by using the *!stringvariablename* format.

Text entered in the following prompts:

First Page	Identifies a link to your HTML document's first page.
Last Page	Identifies a link to your HTML document's last page.
Prior Page	Identifies a link to your HTML document's prior page.
Next Page	Identifies a link to your HTML document's next page.
No Page Selected	Identifies that an HTML page could not be found.
Page Prefix	Identifies a prefix to use with default HTML page numbers.
Go Link	Identifies a link to a default home page that you specify.

Report to HTML Procedure Support

The Report to HTML Procedure template provides exact control as to when to use the output document format, global overrides of selected string settings, and report controls that need to be hidden or skipped when this output format is enabled.

The following options are provided (and only available if the associated Global Extension is active):

Populate on Report

The Report to HTML Global Extension populates this template in *all* reports by default. Select *Never* if you do not want this report to be directed to an HTML document. Select *Always* if you want to have this capability available always for this report at runtime. Select *Conditional* if you need to make this output available based on a certain condition, specified in the next prompt.

Condition

This prompt is available if you have specified *Conditional* in the **Populate on Report** prompt above. A condition can be a simple variable or an expression. If the variable or expression evaluates to non-zero, the HTML document output capability will be active for this report at runtime.

General Tab Options

Output Name Type

If you have directed that the report print to an HTML document file, specify a *Fixed*, *Variable* or *Runtime* name here. *Fixed* will use a default name, and add page numbers and other extensions as discussed on the tab control. *Variable* can be used to designate a fixed name based on a variable condition at runtime, like a date, user name, order number, etc. Use the Fixed or Variable option if you do not plan to have a print preview, and wish to generate the reports directly to HTML.

Runtime allows the end user to specify a custom name and optional folder to direct the HTML document to.

When a report is generated, HTML pages are generated based on the number of pages generated by the report. Hyperlinks to first page, next page, last page, etc. are automatically included. By added a deployment module to your program (i.e., FTP transfer), you now have a way to display all of your Clarion reports as dynamic web pages to your users.

HTML Options

The following prompts allow you to add additional HTML text to your generated output:

Document Name

Enter string text or variable here that will appear on the title bar of your HTML pages. The default text is 'Clarion Report'.

Override Global String Settings

A series of HTML text values are available in the HTML to Report Global Extension for all of your reports. If you need to change specific values for a single report, check this box to enable the **HTML String Settings** button. Press this button to modify the global settings.

Local HTML String Settings

This dialog duplicates the settings available in the HTML Global String Settings dialog. Enter new text if needed here to override the global settings.

Controls Tab Options

The Controls tab displays a list box of every report control's Field equate label that is populated on the report structure. A Field equate label (FEQ) can represent text displayed on the generated report output, but can also identify report logic for Headers, Footers, Detail Bands, Form Bands, and Break structures. Select a control in the list box, and press the Properties button to modify a report's target FEQ.

Report Control's HTML Properties

HIDE (Omit printing)

If a report's FEQ targets a field used specifically for special documents generation (i.e., Hyperlinks, Bookmarks, etc.), check this box to hide (omit) this FEQ from the report's printed output.

SKIP (Omit processing)

A report's contents can be redirected to multiple types of output documents when using the new Advanced Report Generation templates. Check this box to designate that the selected FEQ will not be used for this type of generated document output (e.g., HTML). For example, you would want to skip a control used to generate a special XML tag when generating to HTML.

Target Origin

This prompt is only active if you are not SKIPing the control. For each of the report's FEQs that are active, select *None* from the drop list to designate that the HTML output is plain text information (e.g., the HTML generator will not generate special tag information for this control).

Select *Control Text* to allow the HTML generator to create a hyperlink for this control into the output document. For example, you may have a control in your data table that stores hyperlinks to customer's websites. As the control is generated to HTML, the column's text (i.e. 'http://www.softvelocity.com') will be hypertext.

Select *HardCoded* from the drop list, and press the ellipsis button to select a variable that holds HTML specific information that will be assigned to a report's FEQ that is static (always generated). By contrast, select *Dynamic* from the drop list, and press the ellipsis button to select a variable that holds HTML specific information that will be assigned to a report's FEQ that is conditionally generated. A example of this could be a Detail band that is conditionally generated based on a table column's value. The target variable may hold special HTML tag information that displays color, text, formatting, etc.

Hyperlink Type

This prompt is used to seed the proper hyperlink syntax to the **Target Origin**.

HTTP Adds "*HTTP://*" text in front of the selected **Target Origin**.

MAIL Adds "mailto:" text in front of the selected **Target Origin**.

FTP Adds "FTP://" text in front of the selected **Target Origin**.

NEWS Adds "news:" text in front of the selected **Target Origin**.

Other An additional prompt, **Other Type** appears with this choice. Enter anything into the **Other Type** prompt. This choice is used if you need to add any custom hyperlink that's not included above to your **Target Origin**.

Other Dynamic

An additional prompt, **Other Type,** appears with this choice. Press the ellipsis button to select a variable that will hold the custom hyperlink type information. This choice is used if you need to add any *dynamically changing* or *conditional* hyperlink that's not included above to your **Target Origin**.

None – Adds no additional text to the **Target Origin**

Report to PDF Support Extension

The Report to PDF (Portable Document File) Global Extension template is part of a suite of Advanced Report Generation tools available. This template include the underlying code support to allow you to direct any report's output defined in your application to a PDF document file.

By including this template in the Global section of your application, each report procedure is automatically populated with support for this output type.

The following prompt is provided:

Disable this output

Check this box to make this template inactive in all report procedures. Use this feature when troubleshooting any potential conflicts between other template add-ons and third party products.

See Also: Advanced Report Generation Overview

Report Targets

Report to PDF Procedure Support

Report to PDF Procedure Support

The Report to PDF Procedure template provides exact control as to when to use the output document format, global overrides of selected string settings, and report controls that need to be hidden or skipped when this output format is enabled.

The following options are provided (and only available if the associated Global Extension is active):

Populate on Report

The Report to PDF Global Extension populates this template in *all* reports by default. Select *Never* if you do not want this report to be directed to a PDF document. Select *Always* if you want this output type available always for this report at runtime. Select *Conditional* if you need to make this output available based on a certain condition, specified in the next prompt.

Condition
This prompt is available if you have specified *Conditional* in the **Populate on Report** prompt above. A condition can be a simple variable or an expression. If the variable or expression evaluates to non-zero, the PDF document output capability will be active for this report at runtime.

General Tab Options

Output Name Type
If you have directed that the report print to a PDF document file, specify a *Fixed*, *Variable* or *Runtime* name here. *Fixed* will use a default name. *Variable* can be used to designate a fixed name based on a variable condition at runtime, like a date, user name, order number, etc.

Runtime allows the end user to specify a custom name and optional folder to direct the PDF document to.

When a report is generated, PDF pages are generated based on the number of pages generated by the report. Bookmarks identifying jumps to each page can be designated below.

PDF Options

PDF Options include special features only targeted to PDF document types:

Document Title

Enter string text or variable (using the *!variable* format) to populate the Document Title PDF document tag. This information is usually seen in the title bar of the PDF Viewer.

Application Name

Enter string text or variable (using the *!variable* format) to populate the Application Name PDF document tag. This tag is usually used for other programs to access special information from the PDF through web pages, in this case, the application/PDF name tag.

Program Name

Enter string text or variable (using the *!variable* format) to populate the Program Name PDF document tag. This tag is usually seen used by other programs to access special information from the PDF through web pages, in this case, the programname/PDF name tag.

Subject

Enter string text or variable (using the *!variable* format) to populate the Subject PDF document tag. This tag is usually seen used by other programs to access special information from the PDF through web pages, in this case, the subject/PDF name tag.

Author

Enter string text or variable (using the *!variable* format) to populate the Author PDF document tag. This tag is usually seen used by other programs to access special information from the PDF through web pages, in this case, the author/PDF name tag.

Keywords

Enter string text or variable (using the *!variable* format) to populate the Keywords PDF document tag. This tag is usually seen used by other programs to access special information from the PDF through web pages, in this case, the keywords/PDF name tag.

Use Pages as Parent Bookmark

PDF documents provide bookmarks that are similar to a hyperlinked Table of Contents. Check this box to designate that bookmarks will be created for each report page generated.

Controls Tab Options

The Controls tab displays a list box of every report control's Field equate label that is populated on the report structure. A Field equate label (FEQ) can represent text displayed on the generated report output, but can also identify report logic for Headers, Footers, Detail Bands, Form Bands, and Break structures. Select a control in the list box, and press the Properties button to modify a report's target FEQ.

Controls' PDF Properties

HIDE (Omit printing)

If a report's FEQ targets a field used specifically for special documents generation (i.e., Hyperlinks, Bookmarks, etc.), check this box to hide (omit) this FEQ from the report's printed output.

SKIP (Omit processing)

A report's contents can be redirected to multiple types of output documents when using the new Advanced Report Generation templates. Check this box to designate that the selected FEQ will not be used for this type of generated document output (e.g., PDF). For example, you would want to skip a control used to generate a special XML tag when generating to PDF.

HIDE on PDF (Omit display on PDF)

Check this box to hide the selected control in the output PDF document. This should be used for controls that generate hidden PDF document information, and do not need to be displayed.

Add Field Action

A PDF document is one of the most common document formats used in the office and on the web today. One of the reasons for this is its document versatility. You can embed many types of special document features throughout the document output. Check this box to designate that the selected report FEQ will use one of these special field types.

Field Action

Select *Hyperlink*, *Command Line*, *Annotation*, or *Page Action* from the drop list. This setting will control the Field Action prompts that follow.

Hyperlink Field Action:

Target Origin

This prompt is only active if you are not SKIPing the control.

Select *Control Text* to allow the PDF generator to create a hyperlink for this control into the output document. For example, you may have a control in your data table that stores hyperlinks to customer's websites. As the control is generated to PDF, the column's text (i.e. 'http://www.softvelocity.com') will be hypertext.

Select *HardCoded* from the drop list, and press the ellipsis button on the **Target** prompt to select a variable that holds PDF specific information that will be assigned to a report's FEQ that is static (always generated). By contrast, select *Dynamic* from the drop list, and press the ellipsis button on the **Target** prompt to select a variable that holds PDF specific information that will be assigned to a report's FEQ that is conditionally generated. A example of this could be a Detail band that is conditionally generated based on a table column's value. The target variable may hold special PDF tag information that displays additional color, text, formatting, etc.

Hyperlink Type

This prompt is used to seed the proper hyperlink syntax to the target origin.

HTTP Adds "*HTTP:///*" text in front of the selected **Target Origin**.

MAIL Adds "mailto:" text in front of the selected **Target Origin**.

FTP Adds "FTP://" text in front of the selected **Target Origin**.

NEWS Adds "news:" text in front of the selected **Target Origin**.

Other An additional prompt, **Other Type** appears with this choice. Enter anything into the **Other Type** prompt. This choice is used if you need to add any custom hyperlink that's not included above to your **Target Origin**.

Other Dynamic

An additional prompt, **Other Type,** appears with this choice. Press the ellipsis button to select a variable that will hold the custom hyperlink type information. This choice is used if you need to add any *dynamically changing* or *conditional* hyperlink that's not included above to your **Target Origin**.

None – Adds no additional text to the **Target Origin**

Hyperlink Boxed
Check this box if you would like this hyperlink to have a box drawn around it in the PDF document.

Box Color
This prompt is enabled if the **Hyperlink Boxed** check box is checked. The color that you select here is assigned to the box border generated.

Command Line Field Action prompts:

A PDF document can have hot links that can be used to execute an external program.

Command

Press the ellipsis button to select a valid executable program to run from this control specified in the PDF document output. As always, the program should be in a relative path to the PDF document location.

Parameters

Use this prompt and associated ellipsis button to construct an optional parameter list that can be used with the Command program selected above.

Hyperlink Boxed

Check this box if you would like this command line link to have a box drawn around it in the PDF document.

Box Color

This prompt is enabled if the **Hyperlink Boxed** check box is checked. The color that you select here is assigned to the box border generated.

Annotation Field Action prompts:

A PDF annotation is used to provide additional information in a specific area of the PDF document. You normally double-click on an annotation area and a popup window is displayed with additional information.

Annotation Text Origin

Select *Control Text* to allow the PDF generator to create an annotation mark for this control into the output document.

Select *HardCoded* from the drop list, and press the ellipsis button on the **Target** prompt to select a variable that holds PDF specific information that will be assigned to a report's FEQ that is static (always generated).

By contrast, select *Dynamic* from the drop list, and press the ellipsis button on the **Target** prompt to select a variable that holds PDF specific information that will be assigned to a report's FEQ that is conditionally generated. A example of this could be a Detail band that is conditionally generated based on a table column's value. The target variable may hold special PDF tag information that displays additional color, text, formatting, etc.

Page Action Field Action prompts:

Page Action in a PDF document allows a hot link to other pages in the PDF document.

Action

Select the target location of the page Action from the drop list. These options are self-explanatory.

Hyperlink Boxed

Check this box if you would like this Page Action link to have a box drawn around it in the PDF document.

Box Color

This prompt is enabled if the **Hyperlink Boxed** check box is checked. The color that you select here is assigned to the box border generated.

Add Bookmark

Check this box to allow a report's Field Equate Label's contents to be added to the PDF document's bookmarks

Bookmark Text Origin

Select *Control Text* to allow the PDF generator to create a bookmark based on the control's value into the output document.

Select *HardCoded* from the drop list, and press the ellipsis button on the **Target** prompt to select a variable that holds PDF specific bookmark information that will be assigned to a report's FEQ that is static (always generated).

By contrast, select *Dynamic* from the drop list, and press the ellipsis button on the **Target** prompt to select a variable that holds PDF specific bookmark information that will be assigned to a report's FEQ that is conditionally generated. A example of this could be a Detail band that is conditionally generated based on a table column's value. The target variable may hold special PDF tag information that displays additional color, text, formatting, etc.

Add Bookmark Parent

Check this box to designate that this bookmark will be attached to a parent bookmark that is named below.

Parent Bookmark

Select a parent bookmark that will include the selected control's bookmark in the generated PDF output.

Use PDF Font

Check this box to replace the current font used on the report for this control and replace it with a new name. Select a valid PDF font replacement from the drop list provided.

Report to TXT Support Extension

Report to TXT Support Extension

The Report to TXT Global Extension template is part of a suite of Advanced Report Generation tools available. This template include the underlying code support to allow you to direct any report's output defined in your application to a plain text (TXT) document file.

By including this template in the Global section of your application, each report procedure is automatically populated with support for this output type.

The following prompts are provided:

Disable this output

Check this box to make this template inactive in all report procedures. Use this feature when troubleshooting any potential conflicts between other template add-ons and third party products.

TXT Global String Settings

Press this button to access default text strings that will be used in your plain text output document. More help can be found by clicking on the link below.

See Also: Advanced Report Generation Overview

Report Targets

TXT Global String Settings

Report to TXT Procedure Support

Report to TXT Global String settings

TXT String settings are used to set special characters to represent normally non-printable controls that are populated on a standard Windows report. Global Settings apply to all reports, but you can always override them at the individual procedure level.

Note:

Graphics (IMAGE fields) are NOT supported in TXT documents, and are ignored.

The following tab controls and associated prompts are provided:

Check Box Tab

Checked

If your report uses graphic check boxes, enter a set of valid ASCII characters that will be used to replace a check boxes' checked state. The template default is "[X]"

Unchecked

If your report uses graphic check boxes, enter a set of valid ASCII characters that will be used to replace a check boxes' unchecked state. The template default is "[_]"

Note:

Underscore characters used in the Advanced Report generation templates indicate that a "space" will be output in its position.

Radio Button Tab

Selected

If your report uses graphic radio buttons, enter a set of valid ASCII characters that will be used to replace a radio button's selected state. The template default is "(*)"

Unselected

If your report uses graphic radio buttons, enter a set of valid ASCII characters that will be used to replace a radio button's unselected state. The template default is "(_)"

Lines Tab

If your report uses graphic lines and box controls, you will need to replace these controls with a substitute text character replacement.

On this tab control, the following prompts are presented with their template default:

Drawing Character	*Template Default*
Left Vertical Line	**" | "**
Right Vertical Line	**" | "**
Top Horizontal Line	**" – "**
Bottom Horizontal Line	**" – "**
Left Top Corner	**" / "**
Right Top Corner	**" \ "**
Left Bottom Corner	**" \ "**
Right Bottom Corner	**" / "**

Fill Color Tab

As in graphic lines and boxes, fill colors indeed are not supported in text documents. The templates will substitute each of your colored boxes populated on your report with one of 5 replacement shades. Fill Color 1 represents the lightest shade, and Fill Color 5 represents the darkest shade. The templates will automatically calculate which of the 5 character shades it will replace in your output text document, based on a closest match.

Report to TXT Procedure Support

The Report to TXT (Text File) Procedure template provides exact control as to when to use the output document format, global overrides of selected string settings, and report controls that need to be hidden or skipped when this output format is enabled.

The following options are provided (and only available if the associated Global Extension is active):

Populate on Report

The Report to TXT Global Extension populates this template in *all* reports by default. Select *Never* if you do not want this report to be directed to an TXT document. Select *Always* if you want to have this capability available always for this report at runtime. Select *Conditional* if you need to make this output available based on a certain condition, specified in the next prompt.

Condition

This prompt is available if you have specified *Conditional* in the **Populate on Report** prompt above. A condition can be a simple variable or an expression. If the variable or expression evaluates to non-zero, the TXT document output capability will be active for this report at runtime.

General Tab Options

Output Name Type

If you have directed that the report print to a TXT document file specify a *Fixed*, *Variable* or *Runtime* name here. *Fixed* will use a default name, and add page numbers and other extensions (as discussed on the tab control). *Variable* can be used to designate a fixed name based on a variable condition at runtime, like a date, user name, order number, etc.

Runtime allows the end user to specify a custom name and optional folder to direct the TXT document to.

Save output as separate files

By default, a TXT document will be a continuous single document, regardless of the size of the contents of the generated report. Check this box to designate that TXT documents are generated based on the number of pages generated by the report.

Margin and Page Length

The Margin and Page Length dialog allows you to control the position of the text that will be output to the document file based on a **row and columns** measurement unit.

Use the spin control to set the appropriate **margins** from the document edge. Defaults for all margins is zero (0), which exclude any margins. Maximum margin size is 200.

Initial margin sets the number of rows to skip before generating the first text row. This is only used for the first document page.

Page Length allows you to control how many rows of you report prints on each page. The default is zero (0) which defaults to the report's page length. Maximum page length size (in rows) is 600.

Override global string settings

A series of TXT text values that "translate" graphic controls are available in the TXT to Report - Global String Settingsglobal template. If you need to change specific values for a single report, check this box to enable the **TXT Local String Settings** button. Press this button to modify the default global settings.

Note:

Graphics (IMAGE fields) are NOT supported in TXT documents, and are ignored.

TXT Local String settings

This dialog duplicates the settings available in the TXT Global String Settings dialog. Enter new text if needed here to override the global settings.

Press the **Reset from Global** button to restore the state of the global settings if needed.

Controls Tab Options

The Controls tab displays a list box of every report control's Field equate label that is populated on the report structure. A Field equate label (FEQ) can represent text displayed on the generated report output, but can also identify report logic for Headers, Footers, Detail Bands, Form Bands, and Break structures. Select a control in the list box, and press the Properties button to modify a report's target FEQ.

Controls' TXT Properties

HIDE (Omit printing)

If a report's FEQ targets a field used specifically for special documents generation (i.e., Hyperlinks, Bookmarks, etc.), check this box to hide (omit) this FEQ from the report's printed output.

SKIP (Omit processing)

A report's contents can be redirected to multiple types of output documents when using the new Advanced Report Generation templates. Check this box to designate that the selected FEQ will not be used for this type of generated document output (e.g., TXT). For example, you would want to skip a control used to generate a special XML tag when generating to TXT.

Report to XML Support Extension

Report to XML Support Extension

The Report to XML Global Extension template is part of a suite of Advanced Report Generation tools available. This template include the underlying code support to allow you to direct any report's output defined in your application to an XML (Extensible Markup Language) document file.

By including this template in the Global section of your application, each report procedure is automatically populated with support for this output type.

The following prompt is provided:

Disable this output

Check this box to make this template inactive in all report procedures. Use this feature when troubleshooting any potential conflicts between other template add-ons and third party products.

See Also: Advanced Report Generation Overview

Report Targets

Report to XML Procedure Support

Report to XML Procedure Support

The Report to XML Procedure template provides exact control as to when to use this Extensible Markup Language output document format, global overrides of selected string settings, and report controls that need to be hidden or skipped when this output format is enabled.

The following options are provided (and only available if the associated Global Extension is active):

Populate on Report

The Report to XML Global Extension populates this template in *all* reports by default. Select *Never* if you do not want this report to be directed to an XML document. Select *Always* if you want to have this capability available always for this report at runtime. Select *Conditional* if you need to make this output available based on a certain condition, specified in the next prompt.

Condition

This prompt is available if you have specified *Conditional* in the **Populate on Report** prompt above. A condition can be a simple variable or an expression. If the variable or expression evaluates to non-zero, the XML document output capability will be active for this report at runtime.

General Tab Options

Output Name Type

If you have directed that the report print to an XML document file specify a *Fixed*, *Variable* or *Runtime* name here. *Fixed* will use a default name, and add page numbers and other extensions (as discussed on the tab control). *Variable* can be used to designate a fixed name based on a variable condition at runtime, like a date, user name, order number, etc.

Runtime allows the end user to specify a custom name and optional folder to direct the XML document to.

Save output as separate files

By default, an XML document will be a continuous single document, regardless of the size of the contents of the generated report. Check this box to designate that XML documents are generated based on the number of pages generated by the report.

XML Options

The following prompts allow you to add additional XML text to your generated output:

Each XML document contains one or more elements, the boundaries of which are either delimited by start-tags and end-tags, or, for empty elements, by an empty-element tag. Each element has a type, identified by name, sometimes called its "generic identifier" (GI), and may have a set of attribute specifications. Each attribute specification has a name and a value.

Root Tag Name

A root tag name is required by all XML documents. Enter a hard coded string value, or specify a variable that will hold the root tag name (using the *!variablename* format)

Use Pages as Parent Tag

Check this box to create a parent XML tag for each page generated by the report. These tags are used to navigate from page to page in a browser.

Support Name Spaces

In editing XML documents, it is often convenient to use "white space" (spaces, tabs, and blank lines) to set apart the markup for greater readability. Such white space is typically not intended for inclusion in the delivered version of the document. On the other hand, "significant" white space that should be preserved in the delivered version is common, for example in poetry and source code.

Check this box to enable name spaces support in your output XML document.

Omit XML Header

Check this box to omit the generation of the XML header from your output documents. Some programs that are used to process XML documents may require this.

Which XML format?

This option allows you to designate how the report data generated will be formatted in the XML document. Select from the drop list one of the following supported formats available with the Clarion templates:

Tags – Considered the standard format of XML. Similar to the tag format used in HTML documents. This format is best used when you need to add other XML elements to the report data output.

Note:

By default, the tag format derives its tag information based on the field name (label) defined.

Example:

```
<phone_number>
        <location>Home</location>
        <number>555.555.1111</number>
</phone_number>
```

Attributes

The attributes format is used to simplify data elements that have complex information to output. Using the attributes format is less verbose and easier to use. As with tags, an attribute derives its name from the control name (field equate label) by default.

Example:

```
<phone_number location="home">555.555.1111</phone_number>
```

Metadata Tags

Metadata, loosely defined, is information about information. Additional information about a data element, like a reference URL and author, can be embedded within the XML document.

Note:

By default, the metadata tag format derives its tag information based on the control name (field equate label) defined.

Metadata Attributes

You can also embed metadata information using the shorthand *attributes* format. As with metadata tags, a metadata attribute derives its name from the control name (field equate label) by default.

Refresh values by type

Based on the type of XML format you have selected, press this button to automatically seed all individual report controls with the proper default settings used for the selected format.

Controls Tab Options

The Controls tab displays a list box of every report control's Field equate label that is populated on the report structure. A Field equate label (FEQ) can represent text displayed on the generated report output, but can also identify report logic for Headers, Footers, Detail Bands, Form Bands, and Break structures. Select a control in the list box, and press the Properties button to modify a report's target FEQ.

Controls' XML Properties

HIDE (Omit printing)

If a report's FEQ (**F**ield **Eq**uate Label) targets a field used specifically for special documents generation (i.e., Hyperlinks, Bookmarks, etc.), check this box to hide (omit) this FEQ from the report's printed output.

SKIP (Omit processing)

A report's contents can be redirected to multiple types of output documents when using the new Advanced Report Generation templates. Check this box to designate that the selected FEQ will not be used for this type of generated document output (e.g., XML). For example, you would want to skip a control used to generate a special HTML tag when generating to XML.

Tag Type

There are a variety of XML tags supported and available with the included template. Select from the following tag types:

Tag
These tags are the heart of the XML document. They usually represent data elements that are grouped within a Root Tag Child (required by all XML documents). When Tag is selected as the Tag Type, the **Parent Tag** prompt is enabled and required.

Root Tag Child
The Root Tag Child is a report control FEQ that represents the primary root tag of the output document. There can only be one root tag defined.

Attribute
A report's FEQ can also be used as an XML attribute tag. This is used to assign additional XML attributes to an existing XML tag. Use the **Parent Tag** prompt below to attach the attribute to a target tag.

Value
Select this tag to designate that the report's FEQ is used as "metadata" in the XML document output. Use the **Parent Tag** prompt below to attach the attribute to a target tag.

Html Comment
Select this tag type if the selected control's FEQ will be used as a document comment, similar to a remarks line in programming languages.

Header Attribute
Select this tag type if the selected control's FEQ will be used as an attribute in the XML Header, or document declaration. There are a variety of attribute types supported (e.g., version number, schema name, name spaces settings.)

Parent Tag

This option is enabled when you have selected a standard tag type for the selected report's FEQ. You must specify a parent control that this tag will be assigned to.

Name Type

This option defines the actual name of the tag that will be output to the XML document. From the drop list, select *Control Name* to use the name of the selected control's FEQ with the question mark removed. Select *Control Text* to use the actual contents of the control's use variable as the tag name. Select *HardCoded* to enter string text to use to define the tag name, or *Dynamic* to use the contents of a selected variable to use as the tag name.

Value Type

This option defines the actual value that will be assigned to the XML tag associated with this control. From the drop list, select *Control Name* to use the name of the selected control's FEQ with the question mark removed as the tag value. Select *Control Text* to use the actual contents of the control's use variable as the tag value. Select *HardCoded* to enter string text to use as the tag value, or *Dynamic* to use the contents of a selected variable to use as the tag value.

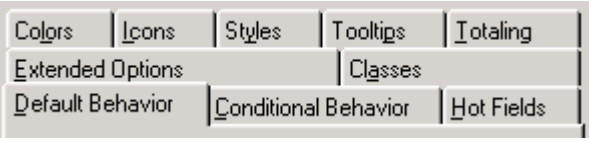
No Value specifies that the tag name contains an empty element, and that the tag name itself is used to format a part the XML document.

Process only once

Check this box to indicate that the document generator should only process the selected control once.

Browse Based Templates

Browse Box control template



The BrowseBox Control template places a "page-loaded" or a "file-loaded" LIST control in a window and generates code to fill the list with data, and to scroll, search, sort, and select the listed items. It generates code to select or filter the data, total the data, update the data directly (edit-in-place), or call a separate procedure to update the data. It also generates code to conditionally set the colors and icons associated with each row and column in the LIST. The standard BrowseBox behavior is defined by the ABC Library's BrowseClass. See *BrowseClass* in the *ABC Library Reference* for more information.

Tip

You can use the BrowseBox Control template to manage a page-loaded *drop*-list simply by setting the DROP attribute to a value greater than zero (0) .

The LIST control's popup menu takes you to the **List Box Formatter** where you can choose which fields or variables populate the list. You can also define how they appear in the list box (including enabling colorization and Icon display). The **Actions** tab on the List Properties dialog provides the prompts which let you define the browse box's functionality, including any record filters, range limits, totaling, scroll bar behavior, and locator behavior.

Placing a BrowseBox on your window

You can place the *BrowseBox* Control template in a window by clicking on the template control tool , then selecting **BrowseBox - File Browsing List Box** in the **Select Control template** dialog. After you select the BrowseBox template, the Application Generator automatically opens the **List Box Formatter** so you can choose the files, fields and variables to display in the list, and you can design the appearance of the list and its fields.

Populating and Formatting the List Fields

The **Populate** button lets you add a field or variable to the list box, one field or variable at a time. The **Select Field** dialog presents the file schematic. Within the schematic, the BrowseBox control appears, with a <To Do> beneath it. To add a field from a data file defined in the dictionary:

1. Select the <To Do> item.
2. Press the **Insert** button
3. Select the file from the **Insert File** dialog.
4. If you want to use a Key, press the **Key** button to select the key from the **Key Access** dialog. If you do not select a Key, the list is displayed in record order, which also disables the ability to set Range Limits.
5. Select a field from the **Fields** list, which appears in the right side of the **Select Field** dialog. After you select the file, key and field (or variable) the **List Field Properties** dialog appears. This lets you precisely define the fields appearance within the list.

Actions

The **Actions** tab of the **List Properties** dialog (right-click the control and choose **Actions**) displays the template prompts which let you specify numerous template options, as well as add custom embedded source code for standard list box events, such as moving the selection bar. The dialog contains the following options:

Default Behavior

Quick-Scan Records

Specifies buffered access behavior for file systems that use multi-record buffers (primarily ASCII, BASIC, and DOS). See *Database Drivers* for more information. These file drivers read a buffer at a time, allowing for fast access. In a multi-user environment these buffers are not 100% trustworthy, because another user may change a record between accesses. As a safeguard, the driver refills the buffers before each record access.

Quick scanning is the normal way to read records for browsing. However, rereading the buffer may provide slightly improved data integrity in some multi-user circumstances at the cost of substantially slower processing.

Loading Method

Select the method used to read the BrowseBox data from the drop-down list.

Page

Page-loading provides near-instantaneous displays for unfiltered data, even for very large datasets. Page-loading uses less memory, because only a few records are held in memory at a time. On the other hand, page-loading can cause erratic vertical scroll bar behavior as well as additional network traffic for each scroll or search action.

File

File-loading provides smooth, accurate vertical scroll bar behavior, plus no additional network traffic when scrolling and searching. File-loading is also quite SQL friendly. On the other hand, file-loading can result in substantial delays and heavy memory usage when reading large datasets.

Accept browse control from Toolbar

Check this box to accept navigation events and other browse control events generated by the *FrameBrowseControl* control template on the APPLICATION's toolbar. See *FrameBrowseControl* for more information on these toolbar buttons and their operation. Clear this box to disable the *FrameBrowseControl* toolbar buttons for this procedure and use local navigation controls only. See also SetToolbarTarget.

LIST Line Height

Enter a number in dialog units (unless PROP:Pixels is active) to set the line height of each row generated in the browse box. This option sets the PROP:LineHeight property. You can optionally press the “E” button to call the Expression Editor, which is used to help you construct syntactically correct expressions or variables, used to calculate the desired LIST Line Height.

Locator

A locator lets the user search for specific records in the list box without manually scrolling through the entire list. **Locator** is only available when browsing a file in Key Order (specify a KEY in the File Schematic). The search field must be the first free key element, that is, the first component field of the browse key that is not range limited to a single value.

For multi-key browses (the Wizards create them), you may have multiple locators. Use the **Conditional Behavior** tab to set additional locators for the additional sorts. Choose from the following locator types in the drop-down list:

None Specifies no locator.

Step Specifies a single-character locator with no locator control required. When the BrowseBox has focus and the user types a character, the list box advances to the first occurrence of the key field beginning with that character (or the next higher character if no keys match the locator character). Retyping the same character advances the list to the next occurrence of the key field beginning with that character.

Use a step locator when the first free key element is a STRING, CSTRING, or PSTRING and you want the search to take place immediately upon the user's keystroke. Step locators are not appropriate for numeric keys. If there is no browse key, the Application Generator converts to no locator.

Entry Specifies a multi-character locator that activates when the locator control is *accepted* (not upon each keystroke). The locator control may be an ENTRY, COMBO, or SPIN. Use an Entry locator when you want to search on numeric or alphanumeric keys, and delay the search until the user accepts the locator control (presses ENTER or TAB). This delayed search reduces network traffic and provides a smoother search in a client-server environment.

The locator control should come *after* the LIST control in the **Set Control Order** dialog.

By default, the locator control is the control whose USE attribute is the first free key element of the browse key. A free component is one that is *not* range limited to a single value. If there is no such control, the Application Generator converts to a Step locator. If there is no browse key, the Application Generator converts to no locator.

When the end user places one or more characters in the locator control, then *accepts* the control by pressing TAB, pressing a locator button, or selecting another control on the screen, the list box advances to the nearest matching record.

Incremental Specifies a multi-character locator, with no locator control required (but strongly recommended). Use an Incremental locator when you want to search on numeric or alphanumeric keys and you want the search to take place immediately upon the user's keystroke.

The locator control should come *after* the LIST control in the **Set Control Order** dialog.

The locator control may be a STRING, ENTRY, COMBO, or SPIN, however, any control other than a STRING causes the Incremental locator to behave like an Entry locator--the search is delayed until the control is accepted.

With a STRING control, when the list has focus, characters are automatically placed in the locator string for each keystroke, and the list box *immediately* advances to the nearest matching record. The backspace key removes characters from the locator string.

We strongly recommend using a STRING control as the Incremental locator control so the search occurs *immediately* with each keystroke, and so the user can *see* the key value for which the BrowseBox is searching.

By default, the locator control is the control whose USE attribute is the first free key element of the browse key. A free component is one that is *not* range limited to a single value. If there is no such control, the Application Generator converts to a Step locator. If there is no browse key, the Application Generator converts to no locator.

Filtered Specifies a multi-character locator, with no locator control required (but strongly recommended). Use a Filter Locator when you want to search on alphanumeric keys and you want to *minimize network traffic*.

This locator is like an Incremental Locator with a record filter. It specifies a *range* of values for which to search and returns a *limited* result set--only those records that fall within the specified range. Each additional (incremental) search character supplied results in a smaller, more refined result set. For example, a search value of 'A' returns all records from 'AA' to 'AZ'; a search value of 'AB' returns all records from 'ABA' to 'ABZ'.

The Filter Locator determines the boundaries for the search based on the user specified search value. The implementation of the boundaries depends on the database driver--for SQL databases, the Filter Locator uses a LIKE; for ISAM databases it supplies upper and lower bounds.

The locator returns *only* the records that match the search value, providing, in effect, a dynamic range limit or filter for the browse.

Tip

The Filter Locator performs very well on SQL databases and on high order key component fields; however, performance may suffer if applied to non-key fields or low order key fields of non-SQL databases.

Override default locator control
The *default* locator control is the control whose USE attribute is the first free key element of the browse key. To override this default and specify a different locator control, check this box. This option is provided in case you have multiple controls with the same free key element as their USE attributes--that is, when you have both ascending and descending keys on the same field.

Select one of the controls to use as the locator control from the **New Locator Control** list.

Find Method
Only available when the Filtered locator is selected. The **Find Anywhere** checkbox determines whether the FilterLocator applies the search value to the entire field (field *contains* search value) or only to the leftmost field positions (field *begins with* search value). If checked, it applies the "contains" test. If unchecked, it applies the "begins with" test. See FloatRight for more information.

Locator Class
Press this button to override the global Locator Manager setting. See Classes Tab.

Record Filter

Type a valid Clarion expression to limit the contents of the browse list to only those records causing the expression to evaluate to true (nonzero or non-blank). The procedure loops through all displayable records to select only those that meet the filter. Filters are generally much slower than Range Limits.

You must BIND any file field, variable, or EQUATE that is used in a filter expression. The **Hot Fields** tab lets you BIND fields.

Press the "E" button to call the Expression Editor. This dialog is used to help you construct syntactically correct expressions to use in the Record Filter prompt.

Range Limit Field

In conjunction with the **Range Limit Type**, specifies a record or group of records for inclusion in the list. Choose a field by pressing the ellipsis (...) button. The range limit is key-dependent. Range Limits are generally much faster than filters.

Range Limit Type

Specifies the type of range limit to apply. Choose one of the following from the drop-down list.

Current Value
Limits the key to the current value of the **Range Limit Field**.

Single Value
Lets you limit the key 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 variables containing the limits in the **Low Limit** and **High Limit** boxes.

File Relationship
Lets you choose a range limiting file from a 1:MANY relationship. This limits the list to display only those child records matching the current record in the parent file. For example, if your list was a list of Orders, you could limit the display to only those orders for the current Customer (in the Customer file).

See also: **Using Range Limits and Filters**

Additional Sort Fields

Specify fields to sort on in *addition* to any Key specified in the **File Schematic** by typing an ORDER expression list (a comma delimited list of field names). See ORDER in the *Language Reference* for more information.

Reset Fields button

Press this button to add Reset Fields. If the value of any Reset Field changes, the procedure refreshes the BrowseBox list. Many BrowseBox events automatically refresh the list; however, if some want another control (such as a Radio button or an entry field) or process to refresh the list, use a Reset Field.

Scroll Bar Behavior button

Pressing this button displays a dialog where you can define the way a scroll bar works.

Choose from **Fixed Thumb** or **Movable Thumb**.

Tip

For file loaded lists, you automatically get Standard Windows standard (movable thumb) scroll bar behavior. However, since this is not possible for page loaded lists, these options let you choose the behavior that best suits your application.

Fixed Thumb

The thumb (square 3D box in the middle of the scroll bar) remains in the center of the scroll bar. CLICK above the thumb to scroll up one "page." CLICK below the thumb to scroll down one "page." DRAG the thumb to the top or bottom of the scroll bar to scroll the top or bottom of the file.

Tip

Choose Fixed Thumb when browsing large SQL tables to get best performance.

Movable Thumb

CLICK and DRAG the thumb to scroll a proportional distance in the list. The thumb remains where you drag it, and its position on the scroll bar indicates the relative position within the browse list.

CLICK above the thumb to scroll up one "page." CLICK below the thumb to scroll down one "page".

When you choose **Movable Thumb**, you can also set the **Key Distribution** to further define how the BrowseBox evaluates the thumb's relative position within the browse list.

Key Distribution

Specifies the distribution of the points of the scroll bar. Choose one of the two predefined distributions (Alpha or Last Names), or Custom, or Run-time from the drop-down list.

Alpha
Defines 100 evenly distributed points alphabetically.

Last Names
Defines 100 points distributed as last names are commonly found in the United States. If the access key is numeric, you should use a custom or run-time distribution.

Custom
Lets you define your own points.

Run-time
Reads the first and last record and computes the values for 100 evenly distributed points in between.

Custom Key Distribution
Lets you specify the break points for distribution along the scroll bar (useful when you have data with a skewed distribution). Insert the values for each point in the list. String constants should be in single quotes (' ').

Run-time Distribution Parameters
Lets you specify the type of characters considered when determining the distribution points. This is only appropriate when the Free Key Element is a STRING or CSTRING. Check the boxes for the types of characters you wish to include for consideration. Choose from **Use alpha characters** (Aa-Zz), **Use numeric characters** (0-9), and **Use other keyboard characters.**

Step Class

Press this button to override the global Step Manager setting. See Classes Tab.

Conditional Behavior

This tab contains a list box that lets you define BrowseBox behavior based on conditions or expressions. Add expressions to the list by pressing the **Insert** button. This displays a dialog where you define the expression and the associated behavior when that expression evaluates to true (nonzero or non-blank).

At run-time the expressions are evaluated, and the behavior for the first true condition in the list is used.

In this dialog you can specify:

Condition	Any valid Clarion expression.
Key to Use	Optionally, the Key to use to sort the BrowseBox data when the expression is true.

The remaining fields and buttons are the same as the **Default Behavior** tab.

Hot Fields

When you select the Hot Fields tab, you can specify fields not populated in the list to add to the QUEUE. When scrolling through the file, the generated source code reads the data for these fields from the QUEUE, rather than from the disk. This speeds up list box updates.

Specifying "Hot" fields also lets you place controls outside the BrowseBox that are updated whenever a different record is selected in the list box. Elements of the Primary Key and the current key are always included in the QUEUE, so they do not need to be inserted in the Hot Field list.

This dialog also lets you BIND a field. You must BIND any file field, variable, or EQUATE that is used in a filter expression.

If the field you are BINDing is not in the VIEW, check the **Not in View** checkbox.

Colors

This tab is only available if you check the **Color Cells** box in the List Box Formatter. It displays a list of the BrowseBox columns that may be colored.

To specify the default colors and any conditional colors, highlight the column's field name, then press the **Properties** button. This opens the **Customize Colors** dialog.

Customize Colors

This dialog lets you specify the default and conditional Foreground and Background colors for normal (unselected) rows; and for selected rows.

Conditional Color Assignments
Below the default colors section is the **Conditional Color Assignments** list. This list lets you set colors to apply when an expression evaluates to true (nonzero or non-blank). To add an expression and its associated colors, press the **Insert** button.

At run-time the expressions are evaluated, and the colors for the first true expression are used.

Icons

This tab is only available if you check the **Icons** box in the List Box Formatter. It displays a list of the BrowseBox columns which can display icons.

To specify default icons and any conditional icons, highlight the column's field name then press the **Properties** button. This opens the **Customize BrowseBox Icons** dialog.

Customize BrowseBox Icons

This dialog lets you specify the default icon and conditional icons for the BrowseBox column.

Default Icon
The default icon to display. Type the icon (.ICO) filename.

You can also name a variable to use as the default icon, using the !*variable* format. The variable may be a string type that stores the icon filename, or check the **The expression is a number** check box if you are referencing a previously named icon assigned to the IconList property.

Conditional Icon Usage
Below the **Default Icon** section is the **Conditional Icon Usage** list. This list lets you set icons to apply when an expression evaluates to true (nonzero or non-blank). To add an expression and its associated icon, press the **Insert** button.

At run-time the expressions are evaluated, and the colors for the first true expression are used.

Styles

This tab is only available if you check the **Style** box in the List Box Formatter. It displays a list of the BrowseBox columns that may have applied styles.

Tip

To specify the default styles and any conditional styles, highlight the column's field name, then press the Properties button. This opens the Customize BrowseBox Styles dialog. A default style may also be defined on the List Box Formatter Appearance tab.

Create GreenBar Effect

Check this box to create a GreenBar effect (alternating colors on each row) on your list box. You will be prompted to select two styles to use that represent the appearance of each alternating row

Alternate Columns

Check this box to apply an alternate style to every other column defined with the style attribute. You will be prompted to select two styles to use that represent the appearance of each alternating column.

Style Type

Use the drop list to select *Local List* or *Style Number*. The *Local List* displays the descriptions of the styles that you created in the Listbox Styles dialog. You can also reference the styles by *Style Number* only.

Style or Style Number

Based on the Style Type selected, select a style's description or number from the drop list control.

Default Style

This entry lets you specify the default style to be used for the column.

Conditional Styles

This list lets you define the styles to apply when an expression evaluates to true (nonzero or non-blank). To add an expression and its associated colors, press the Insert button.

Condition
Provide a valid Clarion expression that when evaluates to true (nonzero or non-blank) will cause the Style to be applied.

Style Type
Use the drop list to select *Local List* or *Style Number*. The *Local List* displays the descriptions of the styles that you created in the Listbox Styles dialog. You can also reference the styles by *Style Number* only.

Style or Style Number
Based on the Style Type selected, select a style's description or number from the drop list control.

At run-time the expressions are evaluated, and the styles for the first true expression are used.

All Styles must be defined. For more information on defining styles see Listbox Styles.

Tooltips

This tab is only active if you check the **Tooltip** box in the List Box Formatter. It displays a list of the BrowseBox columns which may have applied tool tips. Press the **Properties** button to display the *Customize BrowseBox Tooltips* dialog.

Tooltip variable

You can specify a default tool tip string value in the List Box Formatter. This entry lets you override the default tooltip to use a value contained in a variable. Press the ellipsis button to select a variable that will contain the text of your column's tool tip.

Totaling

This tab contains a list box that lets you define total fields for a BrowseBox.

Always Calculate Totaling?
Check this box to always calculate browse totaling fields each time the browse box is refreshed. If this box is unchecked, you can apply a condition to the calculation.

Condition
Enter a variable or expression on this line. If the variable or expression evaluates to a non-zero value, the browse totaling loop will be executed, and all total values updated.

Press the **Insert** button to add total fields. This opens the **Browse Totaling** dialog where you can define total fields for the BrowseBox.

Total Target Field
The variable to store the calculated total. This can be a local, module, or global variable. You may also use a file field; however, you must write the code to update the file.

Total Type

Choose **Count, Sum,** or **Average** from the drop-down list. **Count** tallies the number of records. **Sum** adds the values of the Field to Total. **Average** determines the arithmetic mean of the Field to Total.

Field to Total

The field to sum or average. This box is disabled when the Total Type is **Count**.

Total Based On

Choose **Each Record Read** or **Specified Condition** from the drop-down list. This specifies whether to consider every record or only those that meet the Total Condition criteria.

Total Condition

The condition to meet when using a Total based on a specified condition. You can use any valid Clarion expression. You must BIND any file field, variable, or EQUATE that is used in a filter expression. The **Hot Fields** tab lets you BIND fields.

Extended Options

Disable Auto Size BrowseBox Columns?

If you have enabled the Auto Size Column feature in the Global Properties, check this box if you wish to disable this feature for this procedure's Browse Box.

Disable List Format Manager?

If you have enabled the List Format Manager in the Global Properties' App Settings, check this box to disable this feature

List Format Manager

If the List Format Manager is enabled in the Global Properties, and not disabled, press this button to access the List Format Manager dialog.

SQL Advanced Tab

When the BrowseBox Control template is using an SQL Accelerator driver, SQL Accelerator Drivers convert standard Clarion file I/O statements and function calls into optimized SQL statements, which they send to their backend SQL servers for processing. This means you can use the same Clarion code to access both SQL tables and other file systems such as TopSpeed files. It also means you can use Clarion template generated code with your SQL databases.

This tab control (which only appears when you are using an SQL table) allows you to extend the optimized SQL statements generated by the template.

The following options are available:

Query Elements

The Query Elements items allow you to assign special SQL clauses to your browse box result set. This is accomplished by selecting an existing element in the SQL table, and assigning it to a special function or expression. See Runtime SQL Properties for Views using SQL Drivers

View Field
Select an SQL column from the drop list provided.

Assignment
Enter a valid SQL statement to assign to the View Field.
Example: **`'count(*)'`**

The example above is equivalent to:
"SELECT count(*) FROM *tablename*"

Grouping

Group
To add a GROUP BY clause to the template generated SQL statement, check this box and enter the appropriate SQL string in the **Grouping Definition** field.

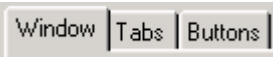
Having
To add a HAVING clause to the template generated SQL statement, check this box and enter the appropriate SQL string in the **Having Definition** field. You must set a GROUP BY definition first to enable the HAVING clause.

See GROUP BY and HAVING and VIEW support for aggregate functions for more information and examples.

Classes Tab

Use the Classes tab to override the global settings for the Class. See Classes Tab.

Browse Customization



The general appearance of the Browse Wizard is controlled by the settings on this window. These settings can be saved in a "theme" for use in future applications.

Procedure Name

Enter a name that the Browse Wizard will use to generate procedure names in your application. The template macro symbol, %FileName, is required, and extracts the Dictionary file name for each file selected by the wizard. You can modify this line with other template macros and text if you wish. No spaces are allowed in procedure names.

Browse Message

Enter default text that the Browse Wizard will generate in the MSG attribute of each list box. The user will see this text in the status bar of the runtime application when the list box is selected. The %FileName macro is not required here, but recommended.

Window

Caption
Text that is entered on this line will be used as the description of the window used by the Browse Wizard.

Secondary Caption
Text that is entered on this line will be used as the message that appears in the status bar of a window only when the browse box is populated as a secondary file on a Form.

Images:
Background
Enter a default image here to use as a graphic or watermark for your Browse window.

Mode
If you have designated a background image to use for your Browse window, this option becomes available to control if the image is tiled, stretched, or centered.

Icon
Enter a default icon to use for your Browse procedure's window. This will allow your Browse window to be minimized if needed.

Font
Press the Font button to select a default font to use for the Browse procedure. Sample text shown below the button is provided to allow you to review your selection.

Options
Select from the drop list to designate the initial position of your Browse window. You can center the window, or use the default position that is set by the template wizard.

In addition, click on the **System Menu** check box to add the Windows System Menu to your Browse procedure. If you will be using any entry fields on the Browse window, you can also click on the **Entry Patterns** check box to allow special formatting picture information (Example: phone numbers or date pictures)

Tabs

Tab Text
Enter text that will be used for each tab control that is generated by the Browse wizard. The default setting is ***&%#) %Key.*** The ampersand (&) identifies the next character as the hot key of the tab control. The **%#** macro identifies the instance or order of the key used in the tab control. The %Key is the description of the key name (or label of the key if the description in the Dictionary Editor is blank).
Example: *2) By Account*

Sort Order Selection Style
Press the **Select Style** button to access the Select Style dialog window.

Buttons

The Buttons tab is the central control for all buttons used on the window. There is a default **Width** and **Height** setting (in dialog units) used for all buttons. However, each button can be overridden individually through the Button Customization dialog.

The default buttons of the Browse wizard are:
Insert
Change
Delete
View
Select
Close
Parent Select
Help

The generation of each button is controlled by other settings within the template wizard. However, you can also override the generation of the **View** and **Help** buttons by unchecking the appropriate boxes.

Browse Procedure Wizard utility template

This wizard creates a multi-keyed Browse Procedure from an existing dictionary file definition. It also creates associated Form (Update) procedures, if you specify that updates be allowed.

There are three models the wizard can use to create procedures: Button, Toolbar, or Both. See Wizard-Control model.

After the introduction screen, you are presented with the following options:

Theme Selection

Theme	Select from the drop list of themes. Themes are groups of settings that control colors, fonts, icons, backgrounds, positions and much more - for Frame, Browse, Form and Report procedures. You will have the opportunity to create a new theme as you progress through the wizard. Select a starting or default theme here.
Save Settings	After you have selected a theme, you have the option to save these settings for any future applications that you create.

What name should be used as the label of the procedure?

Type the browse procedure name.

Which file do you want to browse?

Press the ellipsis (...) button to select a file from the dictionary.

Browse using all record keys

Check this box to make the list sortable on all keys. Clear the box to specify a single sort key.

Allow the user to update records

Check this box to generate a subordinate procedure to update the table. Optionally, provide the name of the update procedure. Clear the box to make the list read only.

Call update using popup menu

Check this box to provide right-click popup menus on the Browse list in addition to any command or toolbar buttons.

Parent Record Selection

This prompt appears only if you specify a single sort key that is the linking key in a Many:One relationship. The Browse Wizard infers from this that you may want to browse only the child records for a specific parent record. Select one of the following to confirm or deny this inference.

Do not select by parent record

Do not limit the browse - in other words, browse all records.

Select parent record via button

Browse only the child records for a specific parent record. Provide a button to select the parent record.

Assume that the parent record is active

Browse only the child records for a specific parent record. Assume the parent record is already active.

Provide buttons for child files

Check this box to provide buttons on the Browse window to access related child tables. Alternatively, related tables may be accessed from the generated update procedure.

Provide a "Select" button

Check this box to provide a "Select" button that displays when the Browse procedure is called to select a record, but is hidden when the Browse is called to update records.

Which control model should the Application use?

Button	The wizard builds the browse with traditional Insert, Change, and Delete command buttons that appear on each dialog.
Toolbar	The wizard builds the browse to use global toolbar command buttons that appear on the application frame. See Control Templates - FrameBrowseControl.
Both	The wizard builds the browse to use both traditional dialog command buttons and global toolbar command buttons.

Customization

Wizards have different "look and feel" settings and actions called *themes*, which can be modified and saved for use in other applications. Themes are set and controlled by a variety of customization options.

Press the **Next** button to accept the selected theme’s settings, or press one of the customization buttons to modify them at this time.

Overwrite existing procedures

Check this box to overwrite existing procedures with the same names. Clear the box to preserve existing procedures.

On the last dialog, the **Finish** button is enabled. If you are satisfied with your answers, press the **Finish** button. You also have the option here to **Save Changes**, where any changes to customization options are saved to the theme that you selected at the start of the wizard. If you wish, you can opt to **Save to a new theme.**

The Browse Procedure Wizard creates the procedure(s) based on the dictionary file and the answers you provided, and then displays the Procedure Properties dialog for your new procedure.

BrowseFuzzyMatching control template

The BrowseFuzzyMatching control template adds a set of controls (a GROUP box containing an ENTRY control and two BUTTON controls, Search and Clear), that lets end users perform weighted searches and sorts of the result.

The query examines the columns in the BrowseBox queue, (ListBox fields and Hot Fields), and returns the data sorted in their order of relevance to the search criteria. At runtime, the BrowseBox may contain an extra column that shows a calculated match result. Prior to searching, the initial match result is 100%. No match to the search criteria gives a 0% match result.

Requirements

This template requires a BrowseBox control.

Populating the Control Template

Open the Window Formatter for any procedure containing a BrowseBox control.

1. **Select Populate ▸ Control Template.**

   The **Select Control Template** dialog displays.

2. **Highlight** *BrowseFuzzyMatching* and press the **Select** button.

3. **Click** on the window at the position where you want the Fuzzy Search controls to be placed.

   This populates an ENTRY control, two BUTTON controls, and a GROUP box surrounding them.

4. **Modify** the text and properties of these controls as desired.

   For example, you may want to change the text of the GROUP box to read "Search Personnel Records" instead of the default text.

5. **RIGHT-CLICK** on the GROUP box, then select **Actions** from the popup menu to set the Fuzzy Match options.

Template Prompts

The BrowseFuzzyMatching template provides the following prompts:

Display Match Results in Browse Listbox
Check this box to display the match result percentage in the Browse Listbox.

Display Match Results in Browse Listbox
Check this box to display the match result percentage in the Browse Listbox.

Display Where
Specifies the position of the match result column in the Browse Listbox. Choose from:

First Column	Populates the match result column as the first column in the Browse Listbox.
Last Column	Populates the match result column as the last column in the Browse Listbox.

Column Format
Specifies the display format of the match result column. The formatting information is set using the same syntax as the FORMAT() attribute for a LIST control.

Match Score Filter Value
Specifies the minimum match value to display in the Browse Listbox. Using a value of 1 will filter out any data that has no matches to the search data.

Reset Fuzzy Match Result Order on Control Event?
Check this box to reset the Browse Listbox results from another control and event on the window.

Control
Choose a control from the droplist that will control the reset of the query.

Event
Choose the event (*Selected* or *Accepted)*, which will force a reset on the query when it occurs on the selected control.

Fuzzy Group Resize Options
Allows the definition of the resizing techniques to use for the group of populated controls.

Resposition Strategy

Specify how the horizontal and vertical position is determined when the end user resizes the window. Valid reposition equates should be used. The reposition EQUATE's are defined in ABRESIZE.INC. The default is set to Resize:FixNearestX + Resize:FixNearestY.

Resize Strategy

Specify how the control's height and width is determined when the end user resizes the window. Valid resize equates should be used. The resize EQUATE's are defined in ABRESIZE.INC. The default is set to Resize:LockSize

Global Options

1. From the Application Tree, press the **Global** button.

2. Select the **General** tab if it is not already selected.

The following prompts are available from the Global Options dialog, **Enable Fuzzy Matching**, and Fuzzy Matching Options—**Ignore Case** and **Word Only**. See *Template Overview—Global ABC Template Settings—General Tab Options*.

BrowseNoRecords Button control template

The BrowseNoRecords button control template provides a standard button that is only available (enabled) when there are records in the target Browse Box control.

If there are no records in the Browse Box control, the button is disabled or hidden.

A popular use for this type of control is a report or process that is called, using the record highlighted in the Browse Box control as a filter criteria, or parent record. Another use of this type of control could be to modify the contents of the Browse Box (marking records, moving them, etc.)

After the button's actions are performed, the browse box can optionally be refreshed.

The BrowseNoRecords button template provides the following prompts:

Select from popup

Check this box to add the button control to the Browse Box popup menu. The text of the button is used as the popup menu item text.

When no Records

Select **Disable** or **Hide** to designate how you would like the button to be displayed when there are no records in the Browse Box control.

Refresh Window After Action?

Check this box to force a window refresh after the button's actions are completed. In the ABC chain, this will generate "ThisWindow.Reset(True)". In the Clarion template chain, this will generate "ForceRefresh = TRUE;DO RefreshWindow".

BrowseNoRecords button control template

The BrowseNoRecords button control template provides a standard button that is only available (enabled) when there are records in the target Browse Box control.

If there are no records in the Browse Box control, the button is disabled or hidden.

A popular use for this type of control is a report or process that is called, using the record highlighted in the Browse Box control as a filter criteria, or parent record. Another use of this type of control could be to modify the contents of the Browse Box (marking records, moving them, etc.)

After the button's actions are performed, the browse box can optionally be refreshed.

The BrowseNoRecords button template provides the following prompts:

Select from popup

Check this box to add the button control to the Browse Box popup menu. The text of the button is used as the popup menu item text.

When no Records

Select **Disable** or **Hide** to designate how you would like the button to be displayed when there are no records in the Browse Box control.

Refresh Window After Action?

Check this box to force a window refresh after the button's actions are completed. In the ABC chain, this will generate "ThisWindow.Reset(True)". In the Clarion template chain, this will generate "ForceRefresh = TRUE;DO RefreshWindow".

BrowsePrintButton

The BrowsePrintButton template provides a **Print** button to call a procedure with an active list box record in memory.

Print All Items

If you use the BrowsePrintButton to call a simple Report procedure, the report prints as usual, applying any design-time keys, sort orders, range-limits, and filters.

Print the Selected Item

If you use the BrowsePrintButton to call a Report procedure with the ExtendProgressWindow extension template (set to Single record), the report reacts to the ProcessRecord request and processes only the selected BrowseBox item. See *Other Templates--ExtendProgressWindow* for more information.

Note: This option works by using a current-value limit on the report key. Therefore, if you have a non-unique key you can print multiple items--for example, all customers named Smith.

The BrowsePrintButton template provides the following prompts:

Procedure name

Type the name of a procedure to call or select a procedure from the drop-down list.

Browse QBEList Control Template

The Browse QBEList Control Template (formerly Soft Velocity Query Center) is a special control template that allows you to test table queries easily and view the results. You can drag columns from a browse box into the query area, enter values, and quickly test your data set.

This template requires that a Browse Box be populated on the window. Although there are no special prompts associated with this template, there are multiple controls that are populated together and are described below:

Filtering Center List Box
This is an EIP (edit-in-place) list box that allows you to enter a column name, operator, Value or Expression, and an optional connecting operator to the next query. You can also drag a column name from the browse box to this list box.
You can also drag specific value from a selected row of the browse. For example, if you select the Column *Name* and drag from a row where the value of that column is *"Bob"*, the QBE List will use that column/value to create *Name = 'Bob'*.

Note:

In the Value or Expression line, do not enter a single quote in your search string. Instead, if you wish to search for ‘FL’ as your search string, enter ‘’FL’’.

Updates to the browse box forces an automatic refresh of the active query.

Case sensitive search for string
Check this box to force case sensitive string searches for the current active query.

Reset Button
Clears the list box for a new query.

Save Query Button
Saves the query (to the non-volatile storage source specified in the application’s global settings).

Note:

Storage is automatic with ABC templates, but you must specify a storage source in the Global Properties of your application when using the Clarion template chain.

Save As Button
Saves the query to a new name that you specify from a query previously loaded.

Load Query Button
Loads a previous query from the non-volatile storage source.

Apply Button
Executes the current active query.

Browse QBE List Template Prompts

The following template prompts are provided:

Filter class

The base class used for the QBE List. The default is **cFilterList**.

Filter object name:

The object name used for this procedure. The default name is **FilterObj**.

Copy generated filter string to clipboard

Check this box to enable the filter string generated by the Filter Class to be copied to the Windows Clipboard. Pressing the Apply button initiates this feature.

Search on string is case sensitive

Check this box to allow the QBE List to default to case sensitive searches.

Case sensitivity search can be set at runtime

Check this box to allow the **Case sensitive search for string** check box to be populated at runtime that allows a user to enable or disable case sensitive string searches.

BrowseQueryButton

The BrowseQueryButton template provides a **Query** button to let the end user apply a dynamic (run-time) filter to the BrowseBox result set. In other words, the end user can query the underlying dataset and display the results of the query in the BrowseBox list.

The default query interface is a dialog with an input field and a comparison operator button for each list box column.

The end user may provide filter criteria for zero or more fields. Additional filter criteria result in a more refined search and a smaller result set (the filter conditions are conjunctive--ANDed together).

Runtime Options

The default comparison operator is (=), which searches for an exact match between the BrowseBox field and the corresponding Query input field. By default all matches are case sensitive. Pressing the comparison operator button cycles through all the available operators:

Operator	**Filter Effect**		
=	*browsefield*	equal	*queryvalue*
>=	*browsefield*	greater than or equal	*queryvalue*
<=	*browsefield*	less than or equal	*queryvalue*
<>	*browsefield*	not equal	*queryvalue*
	no filter		

For string fields, you may use the following special characters in the Query input field to refine your search:

Symbol	**Position**	**Filter Effect**		
^	prefix	caseless (case insensitive) search		
*	prefix	*browsefield*	contains	*queryvalue*
*	suffix	*browsefield*	begins with	*queryvalue*

For example:

d - matches 'd' only
d* - matches 'dog', 'david'
*d - matches 'dog', 'cod'
^*d - matches 'dog', 'cod', 'coD'

Upon completion of the Query dialog, the current sort order of the BrowseBox is filtered to match the query. If Query is selected again, the previous query is available by default. This allows sharing of filters between sort orders, as well as successive filter refinements.

The standard Query behavior is defined by the ABC Library's QueryClass. See *QueryClass, QueryFormClass, QueryFormVisual, QueryListClass, and QueryListVisual* for more information.

The BrowseQueryButton template provides the following prompts:

General

Query Interface

Select the query interface from the drop-down list. Choose from

Form	One input field and button per Query field
List	One listbox row per Query field

Auto Populate

Check this box to provide a query dialog with filter criteria for each field in the BrowseBox. The input fields have the same picture token and prompt as the corresponding BrowseBox field.

Clear this box to enable the **Fields** button and specify custom query input fields.

Caseless Auto Populate

Check this box to provide a query dialog where the filter criteria will not be case sensitive for each field in the BrowseBox.

Fields

Press this button to populate specific query input fields. You can use this option to restrict the query to some subset of BrowseBox fields, or to expand the query to include fields not in the BrowseBox. You can also implement caseless searches by default.

Field	Type the field name to include in the Query dialog, or press the ellipsis button to select the field from the **Select Field** dialog.
Title	Type the prompt or label associated with the Query field.
Picture	Type a picture token for the Query field, or press the ellipsis button to select a token with the **Edit Picture** dialog.
Caseless	Check this box to do case insensitive searches on the Query field. Clear the box to do case sensitive searches.

Disable Begins/Contains?

Check this box to disable Begins/Contains queries. This is available for the QBE form interface for any non-string field.

Retain Query

This option is checked by default, and indicates that the end-user's query will remain in the Query dialog on the subsequent press of the Query button. Clear the check box to reset the Query dialog on each press of the Query button.

Use on startup

Check this box to open the Query dialog before the Browse procedure opens.

Auto-share between tabs

Check this box to make the query applicable to all tabs associated with the browse.

Result Control

Optionally select a STRING Control from the Droplist to display the filter statement created by the QBE object. A property assignment is made to the selected control (using PROP:Text), therefore it is not necessary to associate a variable with the STRING.

QBE Class

Select this tab to override the global Query Manager setting. See *Template Overview--Classes Tab Options--Global* and *Local*.

QBE Visual Class

Select this tab to override the global Query Manager setting. See *Template Overview--Classes Tab Options--Global* and *Local*.

Browse Select button control template

The BrowseSelectButton template provides Select button to choose a record from a list box.

The generated source code gets the currently selected record from the list (makes the selected record the current one in the browsed file's record buffer), and closes down the procedure. For the end user, pressing the Select button is equivalent to double-clicking an item in the list.

The BrowseSelectButton template provides the following prompts:

Hide the Select button when not applicable
: Check this box to hide the Select button when the procedure is not called for selection purposes (GlobalRequest <> SelectRecord).

Allow Select via Popup
: Check this box to allow record selection with a RIGHT-CLICK popup menu. The template adds a popup menu item whose text matches the text on the Select button. The menu item is disabled when the Select button is disabled or hidden.

BrowseToolboxButton template

The BrowseToolboxButton template provides a **Toolbox** button. Pressing the button starts a floating, dockable toolbox containing buttons that invoke the BrowseBox actions defined by the BrowseBox popup menu (Insert, Change, Delete, Select, Print, etc.).

The BrowseBox template automatically adds the Toolbox choice to its popup menu; therefore you can HIDE the **Toolbox** button but still provide access to the toolbox with the popup menu.

The BrowseToolboxButton template provides no configuration prompts.

The standard Toolbox behavior is defined by the ABC Library's PopupClass. See *PopupClass* in the *ABC Library Reference* for more information.

BrowseToolbarControl template

The BrowseToolbarControl template places thirteen (13) standard command buttons on the window. When the user presses these buttons, the template generated code posts appropriate events (scroll up, scroll down, add, change, delete, help, etc.) to the various Browse controls. This template requires a BrowseBox control. See *Control Templates – FrameBrowseControl* for detailed information on each of the thirteen button controls.

Classes

The classes tab lets you control the class (and object) the template uses. You may accept the default Application Builder Class and it's object (reccomended) or you may specify your own or a third party class. 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.

See *Template Overview – Classes Tab Options – Local* for complete information on these options.

Browse Update buttons control template

The BrowseUpdateButtons template provides three buttons for managing file I/O for a BrowseBox: Insert, Change, and Delete. These three button controls act on the records in a browse box. When pressed, the button retrieves the selected record and invokes the respective database action for that record.

The BrowseUpdateButtons template lets you specify an update procedure (recommended for files with two-way relationships) or edit-in-place updates (recommended for lookup files--files with one-way relationships).

The BrowseUpdateButtons template provides the following prompts:

Update Procedure

Type a procedure name or select a procedure name from the drop-down list. If you type a new procedure name, the Application Generator adds the new procedure to the Application Tree.

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)

Use Edit in place

Check this box to let the end user update the browsed file by typing directly into the BrowseBox list. This provides a very direct, intuitive spreadsheet style of update. You may configure the Edit in place behavior with the **Configure Edit in place** button.

Configure Edit in place

Press this button to open the **Configure Edit in place** dialog. This dialog provides the following prompts:

Save

The **Configure Edit in place** dialog offers the **Save** option for four different keyboard actions. These options determine whether changes to an edited record are saved or abandoned upon the following keyboard actions: TAB key at end of row, ENTER key, up or down arrow key, focus loss (changing focus to another control or window, typically with a mouse-click). Choose from:

Default	Save the record as defined in the BrowseClass.Ask method.
Always	Always save the record.
Never	Never save the record, abandon the changes.
Prompted	Ask the end user whether to save or cancel the changes.

Remain editing

The **Configure Edit in place** dialog offers the **Remain editing** option for three different keyboard actions. Check these boxes to continue editing upon the following keyboard actions: TAB key at end of row, ENTER key, up or down arrow key. Clear the boxes to stop editing.

Retain column

The **Configure Edit in place** dialog offers the **Retain column** option for the up and down arrow keys only. Check this box to continue editing within the same list box column in the new row. Clear to continue editing within the left most editable column in the new row.

Insertion Point

The **Configure Edit in place** dialog offers the **Insertion Point** option for initial new record placement in the list. The droplist choices— *before, after,* and *append*— indicate where the edit-in-place row will appear in the list when inserting a record. *Before* and *after* indicate placement in relation to the highlighted record, and *append* places the edit-in-place row at bottom of the list.

Note:

This does not change the sort order. After insertion, the list is resorted and the new record appears in the proper position within the sort sequence.

Column Specific Tab Control

If the Global EIP Configuration is set to *Original*, this interface will appear without the **Class** tab control. The **Class** tab control appears when a *Detailed* Global EIP Configuration is selected.

Select this tab control, then press the **Insert** button to specify the CLASS of object to use when editing a specific list box column.

The **Column Specific** dialog lets you control the class (and object) the procedure uses to edit a specific Browsebox column. You may specify your own or a third party class.

By default, the BrowseUpdateButton template generates code to use the EditEntryClass in the ABC Library. The Global Edit in place (EIP) class must have the EditEntryClass specified as the base class. You can also use the other edit classes or derive your own.

The Application Generator must know about the CLASS you specify--see the *Template Overview--ABC Compliant Classes* section in the *Template Language Reference* for more information.

Column Specific – General Sub Tab
This tab and its prompts within are only available when the Global EIP Configuration is set to *Detailed.*

Allow Edit-In-Place
Check this box to allow edit-in-place for the selected control.

Type
Select from 13 types of window and dialog controls. Each control and dialog support has specific prompts that are applicable to that specific type.

The type control triggers the appropriate EIP class implementation, and options set the properties necessary for that class object. For more information, see the and other supporting classes in the *ABC Library Reference*.

BrowseViewButton

The BrowseViewButton template provides a **View** button that calls the browse update procedure without update functionality (view only). This allows the end user an opportunity to look at a record and its associated child records via the update procedure while not permitting updates to the record.

This template adds a View button and the underlying code to call the update procedure. The code sets GlobalRequest to ViewRecord. The BrowseViewButton template requires an instance of a BrowseBox template.

The BrowseViewButton template provides no additional prompts. It adds embed points for the View BUTTON.

Sort Order Button control template

The Sort Order Button control template is designed to provide a button control that is the central sort order control of a target browse box. When populated, the sort order button automatically hides the tab controls (and optionally, the sheet control), and when pressed, displays a pop up menu of sort order selections.

Sheet Control	Select the Sheet control that contains the tab controls that control the browse box sort orders. Each tab control represents a different browse sorting sequence. The default sheet control Field Equate Label is *?CurrentTab*, but you can select any valid sheet control in the drop list provided.
Hide Sheet	Check this box to hide the sheet (and associated tab controls) that you selected from the **Sheet Control** prompt.

Sort Order Drop List control template

The Sort Order Drop List control template is designed to provide a Drop List control that is the central sort order control of a target browse box. At runtime, the tab controls (and optionally, the sheet control) are hidden, and the drop list displays a list of sort order selections.

Tip

As you initially populate the Drop List control, press **Cancel** when the *Field Selection Dialog* is displayed. Press the **No** Button when you are asked to **Save Properties** in the List Box Formatter. The template will automatically populate the drop list box with sort order choices. The only required prompts are described below.

Another tip; when you populate the drop list, make sure that it is not placed on a tab control, but rather, on the window itself. This will make it visible for all sort choices. You can verify this in the Window Formatter's Property Editor.

Sheet Control Select the Sheet control that contains the tab controls that control the browse box sort orders. Each tab control represents a different browse sorting sequence. The default sheet control Field Equate Label is *?CurrentTab*, but you can select any valid sheet control in the drop list provided.

Hide Sheet Check this box to hide the sheet (and associated tab controls) that you selected from the **Sheet Control** prompt.

Business Rules Templates

Global Support

In Clarion terms, a business rule refers to your business data values stored through the application and to the constraints on attempted changes to those data values.

Business rules are precise statements that describe, constrain, and control selected data elements within your application.

The Global Extension allows you to define rules that can be used throughout your application. In each procedure, you can choose to override all, or part, of these business rules.

The Business Rules Manager template provides the following global prompts:

In the **Mode** group:

Original/Clone selection

The Global Extension operates in one of two modes. In **Original** mode, all of the prompts described in this Help topic appear. When code is generated, the information entered is exported to a file. In **Clone** mode, only the **Mode** prompts appear. The extension imports rules information from the file.

This feature allows you to use the Global Extension in multiple-DLL projects without having to re-enter business rules for each DLL.

Rules File Name

Enter a name for the file, which is exported in **Original** mode and imported in **Clone** mode. The extension for this file will always be **.brf**.

In the **All Rules** group:

The Business Rules Manager template provides to the user at runtime a display window containing a list of rules, and displays any errors made (rules broken) based on the data entered (or not entered).

Default Description

Enter a default description to use for the title bar of this window.

Error-indicator image

Enter an icon image to use that will identify rules errors to the user within the rules list.

Default distance from right

Enter a number in pixels that will separate the error indicator image from the specific rule description.

Global Rules

Press this button to access the Global Rules dialog

View Rules Icons

This group allows you to specify icons to be displayed in the Rules List dialog window. You can designate an icon for the **Header** (top element in a rules tree – normally the Default Description), **Valid Rule** (Rule names that are validated), and **Broken Rule** (Rule names that are not validated).

When any Rule is broken

Regardless of where the Rules Set is defined (Global or Local), the settings here designate specific controls that must be disabled or hidden anytime that a rule is broken. These settings can be overridden on the procedure level.

Disable Form "SaveButton"
Check this box to disable any SaveButton control template when a rule is broken.

Hide Browse "InsertButton"
Check this box to disable any InsertButton that is a populated component of the Browse Update control template when a rule is broken.

Hide Browse "ChangeButton"
Check this box to disable any ChangeButton that is a populated component of the Browse Update control template when a rule is broken.

Hide Browse "DeleteButton"
Check this box to disable any DeleteButton that is a populated component of the Browse Update control template when a rule is broken.

Note:

The specified controls above may be disabled or hidden when the window is first opened, after a rule check reveals one that is broken.

All Rules – Other Controls

Press this button to display a dialog that allows you to hide/unhide or enable/disable special controls anytime that a rule is broken.

Global Rules

The Global Rules dialog window allows you to define unlimited *sets* of Business Rules. Examples might be Global Rules, Customer Rules, Product Rules, Invoice Rules, etc.

Use the appropriate update buttons to add a new set of rules, modify an existing set, or delete an existing set. The following prompts are provided in the update dialog window:

Name

Enter a meaningful name to define your rules set. Example: *GlobalRules*, *CustomerRules*, etc. Note: No spaces are allowed in this name set.

Description

Enter a textual description that describes the rules set. Example: *Customer Information Rules*

In the Global Rules update dialog, you are provided with a list of the specific rules that are contained in the defined set. Use the update buttons to add, modify, or delete these specific rules. The following prompts are provided:

Rule Name

Enter a specific rule name here. The rule name must be in a variable format (e.g. – no spaces, colons and underscore characters permitted, etc.)

Rule Description

Enter a meaningful description of the specific rule that you have defined. Example: "This rule checks if the customer is domestic (CUS:Country = USA)"

Rule Definition – Evaluation Expression

Enter a valid expression that is used as your rules criteria. Example: CUS:Country <> ‘USA’, or CUS:AccountNumber <> 0.

The specific rule that you define will always return a zero if the expression you enter evaluates to "False".

Field to link to

Press the ellipsis button to select a valid field name to link the specific rule to, which will be evaluated whenever this field is populated, or added to the Rules Hot Field list (found in the procedure based Local Rules template).

Display an error indicator when a rule is broken

Check this box to display an error indicator (message) when the specific rule is broken (fails).

Controls Button

Press the Controls button to open a dialog window used to identify a set of generic controls (identified by Field Equate Labels) which will be enabled or unhidden when the rule is evaluated to "True", or disabled or hidden when the rule is evaluated to "False".

Add Standard Procedure Calls

It is sometimes convenient to add code that implements business rules in a procedure. In this case, the rules entered in the Global Rules dialog window merely call the appropriate procedure. To make this structure more convenient to use, the Global Template provides a means by which a standard rules procedure call can be established for every column in a table.

The standard rules procedure call is:

```
MyProc('fieldname',Action,1)
```

Code generated by the Local Business Rules Manager BINDs Action as follows:

```
BIND('Action',ThisWindow.Request)
```

MyProc should be prototyped with three STRING parameters, passed by value, and a LONG return:

```
MyProc PROCEDURE(STRING Field,STRING Action,STRING HandleErrors),LONG
```

More information about standard rules procedures can be found in the *ABC Library Reference* PDF.

The **Add Standard Procedure Calls** dialog provides the following prompts:

File Name

Enter the name of the table for which you want to produce standard rule procedure calls. The ellipsis button can be used to display a list of available tables.

Procedure Name

Enter the name of the procedure to be called. This name corresponds to “MyProc” in the example above.

Do It!

Press the **Do It!** button to generate a rule for every column in the specified table. The template will not overwrite existing rules for a particular column.

Rules - Controls

Enable/Disable Controls

Press the update buttons to add, change, or remove, selected controls that will be automatically enabled if the specific rule that you have defined evaluates to a "True" condition. If the specific rule defined evaluates to a "False" condition, the controls in the list box will be automatically disabled. Enter a Field Equate Label of the control that you expect to evaluate.

Hide/Unhide Controls

Press the update buttons to add, change, or remove, selected controls that will be automatically unhidden if the specific rule that you have defined evaluates to a "True" condition. If the specific rule defined evaluates to a "False" condition, the controls in the list box will be automatically hidden. Enter a Field Equate Label of the control that you expect to evaluate.

Add OK and Cancel Controls

The Add OK and Cancel Controls button populates the appropriate list box with a predefined list of controls.

Local Support

Local Business Rules Manager

The Local Business Rules Manager is used to designate the business rules that will be checked in the designated procedure. The Global Business Rules extension should first be applied before setting the following options:

All Rules Description

Enter the name to be used as the title text of the rules list window. The Business Rules Handler controls the rules list window.

Check Rules After Open Window

Check this box if you wish to check your selected business rules immediately after the window is opened. You can then give the user the option to view "broken rules", so that they are aware of information that must be entered.

Check Rules After Fields Change

Check this box if you wish to check your selected business rules immediately after fields are completed (accepted) on the window. This can allow your business rules to be consistently checked, especially in the scenario where multiple fields are used with a particular rule.

Check for Global Rules Controls

Check this box to apply the global settings of selected controls to apply to this procedure. See Business Rules Controls for more information.

Hot Fields

Press this button to designate special hot fields that may or may not be populated on your window, but are included in the business rules that you need to check.

Used Global Rules

Press this button to allow you to include rule bases (or sets) that you have defined globally.

Local Rules

Press this button to allow you to create new rule bases (or sets) to check locally in this procedure.

Override Global Form/Browse Controls Actions

Check this box to enable the **When any Rule is broken** group, which allows you to override the Global settings.

When any Rule is broken

Regardless of where the Rules Set is defined (Global or Local), the settings here designate specific controls that must be disabled or hidden anytime that a rule is broken. These settings can be overridden on the procedure level.

Disable Form "SaveButton"
Check this box to disable any SaveButton control template when a rule is broken.

Hide Browse "InsertButton"
Check this box to disable any InsertButton that is a populated component of the Browse Update control template when a rule is broken.

Hide Browse "ChangeButton"
Check this box to disable any ChangeButton that is a populated component of the Browse Update control template when a rule is broken.

Hide Browse "DeleteButton"
Check this box to disable any DeleteButton that is a populated component of the Browse Update control template when a rule is broken.

Note:

The specified controls above may be disabled or hidden when the window is first opened, after a rule check reveals one that is broken.

Check for Global All Rules – Other Controls

Check this box to allow this procedure to check for global control equates to hide/unhide or disable/enable when a rule is broken. When unchecked, the global settings are ignored by this procedure.

All Rules – Other Controls

Press this button to display a dialog that allows you to hide/unhide or enable/disable special controls anytime that a rule is broken. On the procedure level, the controls you set are only valid within this procedure's scope.

Local Business Rules

The Local Rules dialog window allows you to define unlimited *sets* of Business Rules. Examples might be Customer Rules, Product Rules, Invoice Rules, etc.

Use the appropriate update buttons to add a new set of rules, modify an existing set, or delete an existing set. The following prompts are provided in the update dialog window:

Name

Enter a meaningful name to define your rules set. Example: *CustomerRules*, etc. Note: No spaces are allowed in this name set.

Description

Enter a textual description that describes the rules set. Example: *Customer Information Rules*

In the Local Rules update dialog, you are provided with a list of the specific rules that are contained in the defined set. Use the update buttons to add, modify, or delete these specific rules.

Local Business Rules - Hot Fields

The Hot Fields dialog window allows you to enter additional fields that may not be populated on the procedure's window, but are a part of some or all of the selected business rules, and need to be included.

Press the Insert button to add a Hot Field, and on the subsequent dialog, press the ellipsis to select a hot field from the Field Selection List.

Local Business Rules - Locally Defined

In the Local Rules update dialog, you are provided with a list of the specific rules that are contained in the defined set. Use the update buttons to add, modify, or delete these specific rules. The following prompts are provided:

Rule Name

Enter a specific rule name here. The rule name must be in a variable format (e.g. – no spaces, colons and underscore characters permitted, etc.)

Rule Description

Enter a meaningful description of the specific rule that you have defined. Example: "This rule checks if the customer is domestic (CUS:Country = USA)"

Rule Definition – Evaluation Expression

Enter a valid expression that is used as your rules criteria. Example: CUS:Country <> 'USA', or CUS:AccountNumber <> 0.

The specific rule that you define will always return a zero if the expression you enter evaluates to "False".

Field to link to

Press the ellipsis button to select a valid field name to link the specific rule to. Whenever this field is populated or added to the Rules Hot Field list (found in the procedure based Local Rules template)

Display an error indicator when a rule is broken

Check this box to display an error indicator (message) when the specific rule is broken (fails).

Controls Button

Press the Controls button to open a dialog window used to identify a set of generic controls (identified by Field Equate Labels) which will be enabled or unhidden when the rule is evaluated to "True", or disabled or hidden when the rule is evaluated to "False".

Local Business Rules - Code Template Error Handler

The Business Rules Handler Code Template is your key to evaluating a defined business rule at any point within your application. Simply include this template at any available embed point, and set the functions and other optional responses when a rule is broken.

Rules to check

Select from **All** (Global and Local), **Global** (Global Only), and **Local** (Local Only) rules to check. By selecting **Global** or **Local**, you can also check an individual rule only.

Rulebase Name

Select a rulebase name to process by the Business Rules Handler. The rulebase names that you see are based on what types you designated in the **Rules to check** prompt

Select Function

The rules handler can perform a variety of functions for you to select from.

Count Broken Rules

Select this function to return a count of broken rules to the designated destination variable.

Destination Variable

Press the ellipsis button to select a variable name that will hold the count of broken rules.

UPDATE() screen before checking rule.

Check this box to force all screen USE variable values to the appropriate memory buffer prior to performing the count function.

Check All rules

Select this function to check all of the business rules designated in the **Rulebase Name**. A Broken rule count will be sent to the Destination Variable.

Destination Variable

Press the ellipsis button to select a variable name that will hold the count of broken rules as all rules are checked

Modify error indicator

When a rule is broken, there are global settings that can be set to handle an rule error that is encountered. Check this box to bypass these global settings.

UPDATE() screen before checking rule.

Check this box to force all screen USE variable values to their appropriate memory buffer prior to performing the rules check function.

DISPLAY() screen after checking rule

Check this box to force a refresh of all screen variables from memory after the rules check is completed.

List Broken Rules
Select this function to display a list of all broken rules defined within the selected **Rulebase Name**. This function assumes that a rules check has already been performed. There are no other secondary prompts associated with this function.

When the list is displayed, you are presented with a "tree" of rules. You can highlight (select) any broken rule, and press the **Go To…** button to jump to the screen variable that the rule is associated with.

Check and List Broken Rules
Select this function to display a list of all broken rules defined within the selected **Rulebase Name**. This function will perform a rules check prior to the display. There are no other secondary prompts associated with this function.

Check and List All Rules
Select this function to display a list of all rules defined within the selected **Rulebase Name**. This function will perform a rules check prior to the display. There are no other secondary prompts associated with this function.

Check a rule

This function is only available when you have selected the Global or Local option in the **Rules to check** prompt.

Destination Variable

Press the ellipsis button to select a variable name that will hold the result of the rule check. A value of one (1) is returned if the rule is broken (e.g., evaluates to "False".)

RuleName

Select a rule name from the drop list to check. This list is filtered based on the Rulebase Name you have chosen.

Modify error indicator
When a rule is broken, there are global settings that can be set to handle an rule error that is encountered. Check this box to bypass these global settings.

UPDATE() screen before checking rule.

Check this box to force all screen USE variable values to their appropriate memory buffer prior to performing the rule check.

DISPLAY() screen after checking rule

Check this box to force a refresh of all screen variables from memory after the rule check is completed.

Business Rules Handler Prompts

The Business Rules Handler Prompts designate specific actions that you wish to take when a rule is evaluated to "True" (passes) or "False" (rule is broken).

Copy to True/False

When you designate a specific action to apply to a rule, press this button to copy the prompt values to the action's counterpart.

Assign Values

Lets you assign values to variables based on the state of the rule check. 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

Lets you specify window controls to hide or unhide based on the on the state of the rule check.. 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.

Enable/Disable Controls

Lets you specify window controls to enable or disable based on the state of the rule check. You may specify multiple controls to enable/disable. Press the Insert button to add a new enable/disable action to the list.

Control to enable/disable

From the drop down list, choose the control to ENABLE or DISABLE.

Enable or Disable control

From the drop down list, choose Enable or Disable.

Display a Message

Lets you specify a message box to display based on the state of the rule check.

Enable Message
Check this box to enable the prompts shown below.

. **Title**

Enter the name of the title to use in the Message window.

Message

Enter the message text.

Icon

Select an optional icon to use in the message box.

Button
Select from the drop list the name of the button that you wish to use for the message box.

Routines/Procedure/Code

Lets you call a selected source code process to execute based on the state of the rule check. You may specify multiple source code calls. Press the Insert button to add a new action to the list

Type

Select from **Routine**, **Procedure**, or **Code**

Routine

Active when Routine type is selected. Enter a valid ROUTINE name here.

Procedure

Active when Procedure type is selected. From the **Procedure** 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. You can also optionally specify a **parameters** list and a **return variable** (as prototyped).

Code

Active when Code type is selected. Enter your custom source code in the text field provided.

Local Business Rules - Special Controls

Enable/Disable Controls

Press the update buttons to add, change, or remove, selected controls that will be automatically enabled if the specific rule that you have defined evaluates to a "True" condition. If the specific rule defined evaluates to a "False" condition, the controls in the list box will be automatically disabled. Enter a Field Equate Label of the control that you expect to evaluate.

Hide/Unhide Controls

Press the update buttons to add, change, or remove, selected controls that will be automatically unhidden if the specific rule that you have defined evaluates to a "True" condition. If the specific rule defined evaluates to a "False" condition, the controls in the list box will be automatically hidden. Enter a Field Equate Label of the control that you expect to evaluate.

Add OK and Cancel Controls

The Add OK and Cancel Controls button populates the appropriate list box with a predefined list of controls.

Local Business Rules - Used Controls

Enable/Disable Controls

Press the update buttons to add, change, or remove, selected controls that will be automatically enabled if the specific rule that you have defined evaluates to a "True" condition. If the specific rule defined evaluates to a "False" condition, the controls in the list box will be automatically disabled. Enter a Field Equate Label of the control that you expect to evaluate.

Hide/Unhide Controls

Press the update buttons to add, change, or remove, selected controls that will be automatically unhidden if the specific rule that you have defined evaluates to a "True" condition. If the specific rule defined evaluates to a "False" condition, the controls in the list box will be automatically hidden. Enter a Field Equate Label of the control that you expect to evaluate.

Add OK and Cancel Controls

The Add OK and Cancel Controls button populates the appropriate list box with a predefined list of controls.

Local Business Rules - Used Global Rules

The Used Global Rules dialog window provide a list of Global Business Rule sets (bases) that you have previously defined in the Business Rules Global Extension.

Select a rulebase name from the list, and press the **Properties** button. The following prompts are available:

Disable this rule?

Check this box to disable the selected global rulebase from the active procedure. Rules that are disabled will not have an icon in the list that identifies them as selected.

Override Display Indicator?

Check this box to override the rulebase display indicator that was defined in the Global Business Rules template

Display an error indicator when a rule is broken

Check this box to display an error indicator (message) when the specific rule is broken (fails).

Use default control if found?

Check this box to use a default control on the window that equates to the current rule defined. If a default control is not found, you can uncheck this box and select one from the subsequent drop list.

Default distance from right

Enter a number in pixels that will separate the error indicator image from the specific rule description.

Controls Button

Press the Controls button to open a dialog window used to identify a set of generic controls (identified by Field Equate Labels) which will be enabled or unhidden when the rule is evaluated to "True", or disabled or hidden when the rule is evaluated to "False".

Form Based Templates

Control Value Validation code template

This code template gets the value of the control and matches it against the value in the key. You can add this code template on an ENTRY, SPIN, LIST, or COMBO control; at the Accepted or Selected embed point. The code generated by this code template gets the value in the control, then matches it against the value in the key.

It can also call a lookup procedure to let the end user select a value. You can check whether the end user has successfully completed the lookup procedure by checking the value of the ThisWindow.Response (or SELF.Response) variable.

Form Wizard utility template

This wizard creates an update Form Procedure from an existing dictionary file definition. The form displays and updates a single record.

After the introduction screen, you are presented with the following options:

Theme Selection

Theme	Select from the drop list of themes. Themes are groups of settings that control colors, fonts, icons, backgrounds, positions and much more - for Frame, Browse, Form and Report procedures. You will have the opportunity to create a new theme as you progress through the wizard. Select a starting or default theme here.
Save Settings	After you have selected a theme, you have the option to save these settings for any future applications that you create.

What name should be used as the label of the form procedure?

Type the procedure name.

Which file do you want the form to update?

Press the ellipsis (...) button to select a file from the dictionary.

Allow Records To Be Added

Check this box to allow new records.

Allow Records To Be Modified

Check this box to allow records to be changed.

Allow Records To Be Deleted

Check this box to allow records to be deleted.

Insert Message

Type the title bar text to display when adding a record.

Change Message

Type the text to display when changing a record.

Delete Message

Type the text to display when deleting a record.

Where do you want this message to be displayed?

Choose the title bar or the status bar.

A field can be displayed that identifies the active record.

Press the ellipsis button to select a column from the dictionary to display on the window title bar.

Validate field values whenever field value changes?

Check this box for immediate validation when the end user "accepts" the column.

Validate field values when the OK button is pressed?

Check this box for column validation on the OK button.

Browsing child files

Select one of the following choices.

Place children on tabs

Access children with push buttons

Do not provide child access

Which control model should the Application use?

Button	The wizard builds the dialogs with traditional Insert, Change, and Delete command buttons.
Toolbar	The wizard builds the form to use global toolbar command buttons that appear on the application frame. See Control Templates - FrameBrowseControl.
Both	The wizard builds the form to use both traditional dialog command buttons and global toolbar command buttons.

Customization

Wizards have different "look and feel" settings and actions called *themes*, which can be modified and saved for use in other applications. Themes are set and controlled by a variety of customization options.

Press the **Next** button to accept the selected theme's settings, or press one of the customization buttons to modify them at this time.

Form Customization

Overwrite existing procedures

Check this box to overwrite existing procedures with the same names. Clear the box to preserve existing procedures.

On the last dialog, the **Finish** button is enabled. If you are satisfied with your answers, press the **Finish** button. You also have the option here to **Save Changes**, where any changes to customization options are saved to the theme that you selected at the start of the wizard. If you wish, you can opt to **Save to a new theme.**

The Form Procedure Wizard creates the procedure(s) based on the dictionary table and the answers you provided, and then displays the Procedure Properties dialog for your new procedure.

Save Button control template

The SaveButton template provides an **OK** button for your window, plus the capability to display an action message for the end user. The SaveButton handles most of the file I/O for the procedure.

The SaveButton template provides the following prompts:

Allow
Check any combination of the three boxes to specify permitted file I/O operations. Conversely, clear the box to prevent the associated operation.

Inserts	Generates code to handle record inserts.
Changes	Generates code to handle record changes.
Deletes	Generates code to handle record deletes.

Tip

The SaveButton template does not detect changes to BLOBs; therefore, if only the BLOB changes, the SaveButton template does not save it.

Field Priming on Insert
Field Priming lets you provide a default value for fields in a new record. This value supersedes any initial value specified in the data dictionary. You can select a field and set an initial value in the **Field Priming** dialog.

Messages and Titles
Press this button to open the **Messages and Titles** dialog to specify update messages and their locations. In addition, this dialog controls some fundamental behavior associated with the procedure, such as whether it confirms before canceling and whether it allows repetitive adds.

View Message
Specifies the text for the action message when the procedure is called to view a record.

Insert Message
Specifies the text for the action message when the procedure is called to add a record.

Change Message
Specifies the text for the action message when the procedure is called to change a record.

Delete Message
Specifies the text for the action message when the procedure is called to delete a record.

On Aborted Add/Change
Specifies the action to take when the user presses the **Cancel** button while adding or modifying a record. Choose from:

> **Offer to save changes**
> Displays a message box prompting to save changes before canceling.
>
> **Confirm Cancel**
> Displays a message box prompting asking if you really want to cancel.
>
> **Cancel without Confirming**
> Displays no message before canceling.

Field History Key
Specify a key that restores the value from the last saved record. When the end user presses the specified key, the generated code restores the field with focus from the previously processed record.

The default key (734) is CTRL+H.

Specifying a key here also enables the FrameBrowseControl's ditto button. This button also restores the value from the last saved record.

When called for Delete
Specify what displays when this procedure is called to delete a record.

Choose from:

> **Standard Warning**
> Displays a message box prompting for confirmation of the delete.
>
> **Show Form**
> Displays the form.
>
> **Automatic Delete**
> Allows records to be deleted without a display or prompt for confirmation.

After successful insert
Select one-at-a-time insert mode or repetitive insert mode.

Choose from:

> **Return to caller**
> Generates a RETURN to the calling procedure following a successful insert. This results in a one-at-a-time insert mode.

Insert another record
Does not generate a RETURN to the calling procedure following a successful insert. This results in a repetitive insert mode.

Ask the user before adding another record
Does not automatically generate a RETURN to the calling procedure following a successful insert, but asks the user whether to add another record.

Location of Message
Specifies where the message displays.

Choose from:

None/Window Control
Embed your own code to display the message in a control.

Title Bar
Display the message in the window's title bar.

Status Bar
Display the message in the window's status bar. Optionally specify which section of the status bar in the **Status Bar Section** box.

Display Record Identifier on the Title Bar
Check this box to append a string to the caption on the window's title bar. Specify the string in the **Record Identifier** field.

Record Identifier
Specifies the string to append to the title bar caption, which you can use to identify the record. Type a string in the Record Identifier box. To use a variable name, precede it with an exclamation point (!).

FormVCRButtons Control Template

The FormVCRButtons Control Template is used to navigate to different records (or rows) from a single Form window. If desired, you can also perform standard database operations on selected records. It is designed as an alternative to the standard Browse-Form paradigm.

The Form VCR buttons control template provides the following functions:

Navigation:

Go to the first record

Page up to the record

Go to the Prior Record

Go to the Next Record

Page Down to the Record

Go to the Last Record

Actions:

Switch form to View-Only mode

Switch the form to Recursive Add Mode. The OK button on the form saves the record and allows you to insert another. The Cancel button cancels the operation and returns to the calling procedure.

Switch the form to change mode. The OK button on the form saves the record and allows you to navigate to another record. The Cancel button cancels the operation and returns to the calling procedure.

Switch the form to delete mode. The OK button on the form saves the record and allows you to navigate to another record. The Cancel button cancels the operation and returns to the calling procedure.

Note:

To make this control template active (or visible) to the Form procedure, you must have a Save Button control template populated.

The FormVCRButtons template is designed to merge the standard functionality of the Browse Box template with a standard form. Moreover, it is actually designed as an *alternative* to the Browse-Form paradigm (e.g., calling a form from a menu).

In addition to these functions, the FormVCRButtons template provides an additional prompts:

Default Behavior

Select from the drop list the default mode to set when the Form window is first opened.

Change to Insert mode if table is empty

Check this box to switch the default behavior to Insert if the primary file specified by the update procedure is empty. The only reason that you would not check this box is the case where Inserts are not allowed on the Form.

Page Size

Enter a number (manually or with the spin control) to designate how many records you wish to Page Up and Down when navigating. The default value is 10.

Tip

If you need to restrict a particular action supported by the FormVCRButtons (i.e., Insert, Change, Delete) you can simply delete the appropriate button.

In order to enable locator activity with the FormVCRButtons template, you must specify a key for the Form's primary file.

The default behavior for this template is the same as a standard Browse Box.

FormVCRButtons – Conditional Behavior

The Conditional Behavior prompts of the FormVCRButtons control template are identical to the Browse Box conditional behavior.

FormVCRButtons – Hot Fields

The Hot Fields prompts of the FormVCRButtons control template are identical to the Browse Box Hot Fields.

FormVCRButtons – Classes

The Classes prompts of the FormVCRButtons control template are identical to the Browse Box Classes options.

Record Validation extension template

This Extension template adds functionality to a Form Procedure template by enforcing data dictionary-defined control value validation. It also lets you specify controls to exclude from validation.

The prompts for this template are accessible through the **Procedure Properties** dialog of a template which includes this extension. A **Record Validation** group box appears in the dialog of the procedure template.

Validate when the control is Accepted
Specifies that validity checking occurs when the control generates an EVENT:Accepted, which occurs when the end user completes or moves the focus from the field.

Validate during NonStop Select
Specifies that validity checking occurs when any control value changes if the window is in AcceptAll (Non-Stop) mode and has focus.

Color Fields rather than Selection
Check this box to change the column or prompt color when an invalid entry occurs.

Field Color when Invalid
Press the ellipsis button (…) to select a color to apply to the column when an invalid entry occurs.

Prompt Color when Invalid
Press the ellipsis button (…) to select a color to apply to prompt when an invalid entry occurs.

Show Message when fields are Invalid
Check this box to display a text message when an entry is invalid.

Message to Display
Specify the text for the message to display when an entry is invalid.

Control to place Message in
Select from the drop down list, the control that will display the invalid entry message.

Do Not Validate
Opens the Do Not Validate dialog, which lets you select fields from a drop down list. The fields you choose will be excluded from validity checks.

SaveChangeButton Control Template

The SaveChangeButton template provides an icon "Save" button  for your form window (or any window that has a Save Button template populated). The purpose of this template is to allow a user to update a form as they navigate from field to field. This can be particularily useful with very large form dialogs.

The SaveButton template provides the following prompt:

Disable Cancel button after Save

Check this box to disable the Cancel button after the user has pressed the SaveChange button. The logical exit from the form after a save is through the OK button (Save Button) only.

ViewFormActions

The ViewFormActions template adds functionality to Form procedures. It gives the ability to configure columns on a window for viewing only. The entire procedure can be set for view only or an individual controls.

The ViewFormActions template provides the following prompts:

Force View OnlyMode
Check this box to put the procedure in view only mode. This generates code that sets GlobalRequest to ViewRecord.

ViewAction
From the drop down list, choose the view action required for each control. Each control has a separate view action.

None	Provides no special attributes to the control.
Hide	Hides the control from view when the procedure is called with a ViewRecord action.
Disable	Disables the control when the procedure is called with a ViewRecord action.
Read Only	Sets PROP:ReadOnly for the control when the procedure is called with a ViewRecord action.
Set Properties	Allows any property to be set for the specific control.
Property	Any runtime property that is valid for the specific control type.
New Value	A value to assign to the control via the property expression.

Tip

For Web applications, Disabled and Read Only ENTRY controls display as text.

Frame Based Templates

Date Time Display extension template

This extension template adds to the functionality of a procedure template, allowing you to display the time and/or date in the status bar, or a control.

The prompts for this template are accessible through the **Procedure Properties** dialog of a template that includes this extension. A **Date and Time Display** button appears in the dialog of the procedure template.

The options that appear in the Date and Time Display dialog are divided into two group boxes, **Date Display** and **Time Display**:

Display in Window
Check the box or boxes to add the date or time display to your window.

Date/Time Picture
Choose a date and/or time display picture from the drop down list. The list displays examples, such as "October 31, 1959," and "5:30P.M."

Other Date/Time Picture
Type in a picture of your choice, if the picture type you wish does not appear in the list. See also: Date Picture Tokens.or Time Picture Tokens

Show the day of the week before the date
(Date only) Optionally displays the day of week.

Location of Date/Time display
Choose between displaying the date and/or time on the status bar, or in a control.

Status Bar Section
When specifying the Date or Time should appear on the status bar, specify the status bar section.

Date/Time Display Control
When specifying the Date or Time should appear in a control, choose the control from a drop down list of field equate labels for the window.

FormVCRControls extension template

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.

Essentially, the *FormVCRControls* Extension provides a "scrolling" Form. You can display, add, delete, or edit many records without returning to the calling Browse to select a new record. However, the keys and filters implemented in the calling Browse procedure do control the navigation of the Form. For example, you can only navigate to records that meet the Browse range limit and filter conditions, and when you navigate to the "next" or "previous" record, the Browse key determines the sequence in which the records appear.

For Form procedures generated by the Application Wizard, if the Form procedure also contains a *BrowseBox*, the *FrameBrowseControl* buttons control the Form when the "form" tab is selected, and they control the *BrowseBox* when the "browsebox" tab is selected. See also *SetToolbarTarget* code template.

FrameBrowseControl

The *FrameBrowseControl* places thirteen (13) standard command buttons on the toolbar of an MDI APPLICATION (*Frame*). When the user presses these buttons, the template generated code posts appropriate events (scroll up, scroll down, add, change, delete, help, etc.) to the active procedure.

Tip

You may delete buttons that your application does not use. For example, the standard templates by default do not use the locate button.

The buttons are designed to work with the *BrowseBox* Control template, the *RelationTree* Control template, and the *FormVCRControls* Extension template; that is, the buttons remain disabled until the program calls a procedure with a *BrowseBox* template or a *RelationTree* template whose **Accept browse control from Toolbar** box is checked, or the *BrowseBox* procedure calls a *Form* procedure with a *FormVCRControls* extension template.

In addition, the called procedure's WINDOW must have the MDI attribute--the standard *Browse* and *Form* templates declare MDI windows by default--you don't need to do anything special to accomplish this. The *BrowseBox* and *RelationTree* templates also checks the **Accept browse control from Toolbar** box by default--so again, you don't need to do anything special to accomplish this.

The *FrameBrowseControl* toolbar buttons operate as follows:

Scrolls to the first row in a BrowseBox or to the previous parent record in a RelationTree. For Form procedures, saves the current record before scrolling.

Scrolls up one page in a BrowseBox or to the previous record on the same level in a RelationTree. For Form procedures, saves the current record before scrolling.

Scrolls up one row in the BrowseBox or to the previous record on any level in a RelationTree. For Form procedures, saves the current record before scrolling.

Locates a specific record in a BrowseBox. See *Control Templates--BrowseBox* for information on specifying these locators.

Scrolls down one row in the BrowseBox or to the next record on any level in a RelationTree, expanding the tree branch if necessary. For Form procedures, saves the current record before scrolling.

Scrolls down one page in the BrowseBox or to the next record on the same level in a RelationTree. For Form procedures, saves the current record before scrolling.

Scrolls to the last row in the BrowseBox or to the next parent record in a RelationTree. For Form procedures, saves the current record before scrolling.

Selects the highlighted row in a BrowseBox.This is only appropriate when the Procedure is called to select a record. For example, when called as a lookup.

For a BrowseBox, calls a *Form* procedure to add a new record. For a RelationTree, calls a *Form* procedure to add a child record of the currently highlighted record. For a Form procedure, adds another record of the same type.

Calls a *Form* procedure to change the record highlighted in the BrowseBox or RelationTree.

Deletes the record highlighted in the BrowseBox or RelationTree. The BrowseBox delete behavior is determined by the settings on the Update Buttons Control template.

On a *Form* procedure only, pastes into the field with focus, the corresponding value from the previously processed record (the value in the record buffer). In other words, repeat the value from the previous saved record.

Invokes Windows standard help behavior: calls WINHELP.EXE with the help topic or keyword specified by the WINDOW's HLP attribute.

The Properties dialogs for the *FrameBrowseControl* buttons is the normal **Button Properties** dialog.

Graphing

Getting started with the Graph Control template

The new Graph Control template provided in this release offers a wide variety of powerful features and template based options. Even to an experienced user of Graphing controls and products, the template prompts may look a bit complex when using the Graph control for the first time.

This topic is designed to give you a step-by-step look at developing a graph control in your windows and reports, with a brief explanation of the terminology used in the Clarion Graph Control template.

Design Steps Checklist

1. Make sure that SVGraph templates are registered. Check your Template Registry to confirm this.

2. Add the SVGraph Global Extension to the Global section of your target application.

3. Populate the target window or report with the graph control. The Graph control template should be in the list of control templates available to populate.

4. Determine which view of the graph you want to see and what features it should have.

   Look at the sample graph types and parts of a graph topics to help you decide.

5. Fine-tune your graph's accessory options. To do this, start with the *General* **View** of the Graph Control template.

   If you want to provide a popup menu for your users, check the "PopUp menu" checkbox.

   If you want to provide Tool Tips on your graph, check the "ToolTip" checkbox.

   If you want to provide Drill Down support on your graph, check the "Drill Down" checkbox.

   Of course, the options mentioned above do not apply to graphs populated on reports.

6. Change the **View** option to *Initial values*. Change any default parameters if necessary.

7. Change the **View** option to *Display options*. Change any default parameters if necessary.

8. Determine what data you want to display on the graph, and identify the source where the data will be extracted. Your options are **File** and **Queue**. If the data to be graphed is read from a file, then add the appropriate file in the *Table Schematic* window (You should see a "To Do" on the Graph Control item).

9. Prepare your data for display on the graph. To do this, Change the **View** option to *Data source.* Press the **Insert** button to add your data source.

   Select the source of the data (*File* or *Queue*). Select an alternate key or filter criteria if necessary.

10. Navigate to the **Points** tab. Add the necessary amount of data series that you need, and adjust each series if necessary.

    Note:

    Each series will be displayed as a separate diagram (drawing) on the composite graph.

    Press the **OK** button to save your data source and associated series points.

11. Compile and test, and repeat any of the above steps where necessary.

Of course, the above steps do not cover all of the options and features available, but is designed to help you get started. Refer to the shipping graph example and on line help for more information.

SVGraphGlobal - Graph Global Extension

The SVGraphGlobal extension template is a requirement for any application that will utilize Clarion's graphing support. This extension simply includes the necessary equates and Classes needed to compile the application.

Requirements

This global extension template has no template requirements.

Template Prompts

The SVGraphGlobal template extension has no template prompts.

The ABC Library file button is used to designate the SVGraph base class. The default is *GraphClass*.

Graph - Graph Control

The Graph Control template is the definitive tool that is designed to solve nearly all of your business graphing needs.

Due to the complexity and options needed for this template, a drop list divides the template prompts into five (5) distinct categories:

General *Initial values* *Data Sources* *Display Options* *Classes*

General

Refresh Automatically

Check this box to direct the templates to generate the appropriate graph refresh each time the window is refreshed, or in reports, each time a record is read. You can also control the refresh of the graph with the appropriate Graph Refresh Button or Code template.

Accessibility

Accessibility determines how much access or control you want to give to the user to control the graphing display. Select from the following choices.

Popup Menu
Check this box to create a popup menu that allows the user to control various elements of the graph display. This popup is available when the user RIGHT-CLICKS over the graph control. These elements are controlled in the Display options area.

Tooltip
Check this box to allow a tool tip to be displayed when the user moves the mouse over the graph control. The amount of information displayed is controlled in the Display options area.

Drill Down
Check this box to allow the user to DOUBLE-CLICK on a graph control to generate a call to the template's DRILLDOWN procedure. Normally, you use this to generate an additional graph based on values extracted from the area of the graph control that was clicked on.

Additional settings and design information of the Drill Down technique is located in the Display options **View** selection.

Initial Values

This area of the Graph Control sets the initial display state of the graph when first loaded and viewed by the user. Keep in mind that these values are only applied when the graph control is active and displayed for the first time. The Graph Control template provides additional user control of a graph's style, type, scale, etc. at runtime.

The following options are provided:

Graph Tab Options

Graph Type
Select an initial graph type to display. For a quick look at the different graph types supported, click here.

Graph subtype
This drop list is available for accumulator graph types (Bar and Column). Choose from simple, which displays bars and cylinders of variable length, or normalized, which makes all bars and columns equal height, and normalizes each node value accordingly.

Graph figure
Select the type of lines drawn for the Graph Type selected will be a *bar* (rectangular) or *cylinder* (circular)

3D-effect
Select *Yes* from the drop list if you wish to enable a three-dimensional graph effect when the control is first loaded.

Title Tab Options

The title simply refers to a textual string that displays at the top of the graph control.

Title
Select *Yes* from the drop list if you wish to display a title when the graph control is first loaded.

Properties – Title Text
Enter a string to display as the graph title. You can also press the ellipsis button to select a variable's contents to use for the graph title. Make sure to place an explanation point in front of the variable name (i.e., *!LOC:Title*). Press the **F** button to set an initial font and style for the graph title.

Legend Tab Options

This option controls a graph's guide or key that helps the user understand what they are looking at.

Location
From the drop list, select the location that the legend will appear in relation to the graph. If you do not wish to display a legend, select *None* from the drop list.

Box
Select *Yes* from the drop list if you want to surround the legend with a box.

Axis Tab Options

The Axis tab specifies the line drawing and appearance of the X and Y axis of the graph control displayed.

General

The template prompts on this tab apply to attributes shared by both the X and Y axis.

Axis styles
Axis styles define the lines and types of measurement intervals that you can display around the graph type selected. Select *None* if you do not wish to display any X and Y-axis lines. Select Standard to display the X and Y-axis with a maximum of measurement interval (tick marks) for accurate graph reading. Select *Long* to display the X and Y-axis with a minimum of measurement intervals for reading the graph.

Name
You can define a name to use for both the X and Y-axis in their respective tab controls. This prompt gives you the ability to disable both of them when the graph is initially loaded. The user can use the popup menu feature (if enabled) to allow the axis names to be displayed when needed. Select *Yes* to display any defined axis names when the graph control is initially loaded. Select *No* to hide the names when the graph control is initially loaded.

Grid
Grid lines are dotted lines that are overlaid on top of the graph control. You have the option in the respective **X-Axis** and **Y-Axis** tab controls to enable them. In this prompt, select *No* to disable all grid lines when the graph is first loaded, or select *Yes* to display any grid lines that are enabled in the X-Axis and Y-Axis tab controls

Scale Style
Select *Linear* from the drop list to specify equal increments of scaling to use in your graph control based on the highest maximum data point value. Select *MS-Word* to define a scaling technique used by MS-Word that rounds off axis values. This type of scaling should be used when a maximum value can cause linear scaling to give a skewed or distorted graph presentation.

Scale min/max
Select *Yes* to activate graph scaling based on minimum and maximum data point values. The graph minimum scale will be based on the lowest data point value (i.e., instead of zero). This technique is useful for an optimum graph display when all data point values fall within a relative contiguous range.

X-Axis (Horizontal Properties)

This tab control provides the initial values for X-Axis properties. Keep in mind that many of the following settings are enabled or disabled by the General sub tab discussed above:

X-Axis Label

Press the ellipsis button to select a variable to use as the X-Axis Label. This label is displayed on the graph control's x-axis.

Format of values

Press the ellipsis button to build a picture using the *Edit Picture String* dialog, or manually enter a valid picture that will be used to format the numeric elements displayed on this axis.

Name
Enter a string value, or press the ellipsis button to select a variable to use as the description of this axis. Press the F button to specify a special font and font size to use. The description is used on the tool tips or target controls that are activated when the mouse moves over the axis label area.

Grid

Grid lines are dotted lines that are overlaid on top of the graph control. In this prompt, select *No* to disable all vertical grid lines when the graph is first loaded, or select *Yes* to display any vertical grid lines.

Color

Press the ellipsis button to select a color to use for the X-Axis line.

Thickness

Enter a thickness in dialog units to use for the X-Axis line.

Scale interval

Enter a numeric value to designate the "hash mark" intervals that will be displayed on the X-Axis.

Y-Axis (Vertical Properties)

This tab control provides the initial values for Y-Axis properties. Keep in mind that many of the following settings can be enabled or disabled by the General sub tab discussed above:

Y-Axis Label

Press the ellipsis button to select a variable to use as the Y-Axis Label. This label is displayed on the graph control's y-axis.

Format of values

Press the ellipsis button to build a picture using the *Edit Picture String* dialog, or manually enter a valid picture that will be used to format the numeric elements displayed on this axis.

Name

Enter a string value, or press the ellipsis button to select a variable to use as the description of this axis. Press the F button to specify a special font and font size to use. The description is used on the tool tips or target controls that are activated when the mouse moves over the axis label area.

Grid

Grid lines are dotted lines that are overlaid on top of the graph control. In this prompt, select *No* to disable all horizontal grid lines when the graph is first loaded, or select *Yes* to display any horizontal grid lines.

Note:

Grid lines are also dependant on the graph type. For example, if you are using a bar graph style with a certain thickness, the grid lines may be hidden.

Color

Press the ellipsis button to select a color to use for the Y-Axis line.

Thickness

Enter a thickness in dialog units to use for the Y-Axis line.

Scale interval

Enter a numeric value to designate the "hash mark" intervals that will be displayed on the X-Axis.

Grouping

Sort

If your data points are grouped together, this option specifies whether to group by *Name* type (the data points of the graph are alphanumeric) or by *ID* (the data point of the graph should be numerically sorted).

For example, if a data point is a string type containing numeric data, selecting *Name* will sort the data like this:

```
1, 101, 1001, 20, 203, 2033
```

where an *ID* sort will sort the values above as follows:

```
1, 20, 101, 203, 1001, 2033
```

Sequence of Sorting

Specify that the Initial Value Sort Grouping selected above will be in *Ascending* or *Descending* sequence.

Node Tab Options

Essentially, a graph's architecture is composed of a finite set of dots called **nodes,** connected by links called **edges**. This section focuses on the node parameters of your graph.

Dependant on the graph type used, nodes determine how a graph is rendered, and identifies key data element values used to construct the graph.

Node type

Select how a node will be represented. Choose *Triangle*, *Square*, or *Circle* from the drop list. If you wish to hide the nodes in your graph, select *None*.

Node radius

Enter a numeric value manually or by spin control that will determine the size of a radius of a node point in dialog units. If omitted, the default size is 3.

Minimum/Maximum

Select *Yes* if you wish to identify Maximum and Minimum node values in your graph. These special nodes will appear slightly larger than the other nodes.

Text of node

Each of your graph node points can have an assortment of values assigned to them.

Show Label on Nodes

Select *Yes* if you would like the data point labels to be displayed at each node point.

Note:

The Graph type you select will control whether or not a Node label can be displayed. For example, a bar graph can hide a node label, where a line graph will not.

Show Value on Nodes

Select *Yes* if you would like the data point values to be displayed at each node point.

Background

Select *Yes* if you would like the label and/or value to display its background color defined. If you would like these text fields to have a transparent background, select *No*.

Best position

Select *Yes* if you would like the template to calculate the best position for your node label or value. If you select *No*, the node label or value will always appear to the upper right of the node point, and some values may be clipped from the graph's display.

Other tab options

Zoom

Select the initial zoom setting when the graph control is first displayed. The *Auto* setting will zoom to a "best fit" sizing.

ToolTip

Select *Yes* from the drop list to automatically display tool tips when the graph control is first displayed.

Best Position

Select Yes from the drop list to allow the tool tips to "float" as you drag your mouse near the edge of the graph control. Select No if you want the tool tip display to be cropped or truncated as you move the mouse to the edge of the graph control.

Indent

Indents for drawing

The Indent tab sets margins from the edge of the graph control where the graph data will be written. Values are in dialog units. Use the spin controls to set the margins for **Top**, **Bottom**, **Left**, and **Right**.

Print

The Graph control template has powerful built-in support for printing your graphs. The following initial settings are available.

Preview

Select *Yes* to call the Print Preview window prior to printing your Graph control.

Orientation

Select *Portrait*, *Landscape*, or *As on Printer* (printer default) from the drop list to designate how the graph control will be printed on your paper.

Indents from edges of a paper (mm)

Specify the non-printable area of your printed output by setting the **Top**, **Bottom**, **Left**, and **Right** indents in millimeters.

Box

Select *Yes* if you would like to print a box (border) around the printed graph control.

Other

Wallpaper

Select *Yes* from the drop list to display a graphic image when the graph control is first displayed. Press the ellipsis button to select a valid graphic image.

Save

Press the ellipsis button to select a variable that will hold the name of a Windows Meta File (*.WMF) that will store the graph control's contents. This file is written when the user selects "Save" from the popup menu. You can also enter a string filename (Example: 'mygraph.wmf')

Data sources

Data sources identify the source of the values that are plotted for the graph control. The prompts that follow are the heart of this powerful template. Multiple sources are possible. Press the **Insert** button to add a new data source, **Properties** to examine an existing data source, or **Delete** to remove a data source.

The following template prompts are provided:

Data Source tab options

Source Name

Enter a meaningful description that will uniquely identify the data source. You can enter up to 255 characters.

Source

Select *File* if the data values used for the Graph Control will be extracted from a data table. Select *Queue* if the data points will be extracted from a global or locally defined QUEUE.

Data From File

The following prompts will be available when you select File as the Data **Source**.

This section simply controls how the file will be processed. As the file is processed, there will be field values in the records that are read that will be used as the Points used for the graph control.

Key to Use

Press the ellipsis button to select an optional key that will be used to process the file in a specific sequence.

Record Filter

Type a valid Clarion expression to limit the contents of the processed data source to only those records causing the expression to evaluate to true (nonzero or non-blank).

Range Limit Field

In conjunction with the **Range Limit Type**, specifies a record or group of records for inclusion in the list. Choose a field by pressing the ellipsis (...) button. The range limit is key-dependent. Range Limits are generally much faster than filters.

Range Limit Type
Specifies the type of range limit to apply. Choose one of the following from the drop-down list.

Current Value
Limits the key to the current value of the **Range Limit Field**.

Single Value
Lets you limit the key 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 variables containing the limits in the **Low Limit** and **High Limit** boxes.

File Relationship
Lets you choose a **Range Limiting File** from a 1:MANY relationship. This limits the list to display only those child records matching the current record in the parent file. For example, if your list was a list of Orders, you could limit the display to only those orders for the current Customer (in the Customer file).

SQL Filter
When the active data source is using an SQL Accelerator driver, SQL Accelerator Drivers convert standard Clarion file I/O statements and function calls into optimized SQL statements, which they send to their backend SQL servers for processing. If you would like to filter an SQL table using SQL statements, press the ellipsis button to select a variable that will hold the SQL statement.

You can also manually enter an SQL filter expression into this prompt.

Additional Sort Fields
The purpose of this group is to add additional sort fields to the default sequence that the data source is processed. This sequence is applied to the data source's default sort behavior. If no default sort is specified (no key), the additional sort fields' dialog sets the primary sort.

Press the appropriate update buttons to add, change or remove a sort field. On the **Field** prompt, press the ellipsis button to select a field name from the Select Column dialog. In the **Sequence** drop list, select *Ascending* or *Descending* to designate the sort fields' sort sequence.

Data From Queue
The following prompts will be available when you select *Queue* as the Data **Source**.

Queue Name

Press the ellipsis button to select a valid Queue label from the *Select Column* dialog. At press time, an exclamation point is automatically prepended to the name that you select.

Sort Fields

Sort Fields designate the order that the queue will be read by the data source. This improves the internal processing of the graph's data points. Press the appropriate update button to add, change or remove a queue's sort field. On the subsequent *Sort* dialog, add each **Sort** field by pressing the ellipsis, then select *Ascending* or *Descending* from the **Sequence** drop list.

Record Filter

Press the ellipsis button to select a variable to use in a filter expression for the processed queue. This is usually a Boolean (True/False) condition used to reject certain queue records. You can also manually enter a filter expression in this field.

Points tab options

Each point that you create in this section becomes a reference point, or node, for your graph control. How these points are displayed depend on a number of factors, including graph type, initial settings, etc. It is important to establish your data source first before attempting to set any data points.

Series Name
Enter a descriptive name for your point "series". A series represents a collection of data values read from a file (table) or queue and consolidated to a single graph result. This description is used on the graph's legend and node to describe this data point.

Note:

"Series Name" is synonymous with "Diagram Name".

Filling

The **Filling** button controls how each data series is displayed. This display is dependent on many factors, including graph type (bar, line, etc.) and initial settings. The following prompts are provided:

Style

Currently, *Solid* is the only fill style that you can select.

Type

This refers to the color type. You can specify a *Fixed* , *Variable,* or a *Default* color type. Default generates a specific color determined internally by the internal class methods referenced by the templates.

Color

If you have selected a *Fixed* color type, press the ellipsis to select a color from the *Color* dialog. If you have selected a *Variable* color type, press the ellipsis to select a variable that will store a valid color equate.

Grouping of point

Grouping of point designates that the active point will be displayed together with other points that name the same group. Select *Yes* to group this data point. Select *No* if this data point will be separate and distinct from other data points.

Tip

Select *No* here if your graph type will be a linear type (line graph scatter graph, etc.). Grouping of points are best used with Bar and Column graph types.

Name

Press the ellipsis button to select a variable name that this data point will associate with. Different data points that you need to group together should have the same variable value (i.e., job hours grouped by department name – select !*DEP:name* here.)

Text

Press the ellipsis button to select a variable name that this data point will associated with. The contents of this variable will be displayed along with the data point (node) on the graph control. Your graph type selected and Initial settings may dictate how this value is displayed.

Record Action

Select from the drop list how each record read (set by the Data Source parameters) will be processed into the data point.

If you would like every data value read to be displayed separately on your graph control, select *Graph* from the drop list.

If you would like every value read by the data source to be counted together into a data point, select *Count* from the drop list.

If you need a summation (addition) of every value read by the data source, select *Sum* from the drop list.

If you would like every value read by the data source to be averaged together into a single data point, select *Average* from the drop list.

Field

Press the ellipsis button to select a variable that will be used to determine this data point(or node) value. For example, *!EMP:WorkingHours* will display the employee's working hours in the form of whatever Record Action that is selected above.

Note:

Make sure that your **Field** selected is a numeric data type.

Point Filter

Enter an optional expression here that can be used to filter the data point. This is similar to a detail filter on a report procedure. The Data Source tab control can set initial filtering as necessary, and the individual data point can be processed further here.

Field name

Press the ellipsis button to select a variable that will identify this point (or node) on the graph control. For example, *!DEP:Name* will display each department's name at the node position. Node values can be displayed by tool tip, or statically displayed on the graph.

Break

Breaks are essential for all types of **Record Action**, with the exception of the Graph action (which is essentially no action). A break is used to generate different data point nodes anytime the break value changes.

Use a break to stop a Sum, Count, or Average, and start a new one. For example, if I needed to tally the hours worked for each department on a particular project, I would add the EMPJ:DepartmentId field as a Break Field.

Use the appropriate update buttons to add, change, or remove a break field variable. On the *Break Fields* dialog, press the ellipsis to select a break field for the Field prompt.

Hot Fields tab options

This tab control is only available when the **Data Source** selected is from a *File* (Table).

Press the appropriate buttons to add, change and remove Hot Fields. On the subsequent *Hot Fields* dialog, press the ellipsis button to select a variable to use in the **Field** prompt. Optionally check the **BIND Only** check box if the variable you select is not part of a Table that has the BINDABLE attribute set from the data dictionary.

Any variable used in an expression to calculate the break points of a Sum, Count, or Average **Record Action** must be identified here as a Hot Field.

Display options

The *Display options* **View** selection provides the necessary prompts to control how much user control you need to provide for the Graph control. In addition, this area allows you to extend the functionality of your Graph Control by extending Drill Down capabilities and enhancing static display elements of your graph (e.g., display items that can not be modified by the user).

The following template prompts are provided:

PopUp Menu tab options

The **PopUp** tab control contains settings for a very powerful option that provides your users with a popup menu of graph options at runtime. With the number of popup options available, the user can change nearly every aspect of the Graph control. The best news of all is that it is very easy to implement.

To enable the options shown on the **Popup** tab control, make sure to check the PopUp Menu check box found in the Accessibility options of the General View.

General

The popup selections in the **General** tab are menu items that the user will first see when right-clicking over the graph control. Popup Items marked in **bold** provide access to sub menu options, shown on other tab controls here if the target check box is checked.

For an additional description of each menu option shown here, please refer to the General and Initial Settings **View** options.

Graph Type

The menu options displayed here are only available if the **Graph Type…** check box is checked on the **General** tab.

At press time, there are 11 different graph types that are available in the Graph Control. Care should be taken to give your user only those graph types that will provide a meaningful display of the data elements you have specified.

Legend

The menu options displayed here are only available if the **Legend…** check box is checked on the **General** tab.

Legend options are described in detail in the Initial Settings View selection.

Axis

The menu options displayed here are only available if the **Axis…** check box is checked on the **General** tab.

Axis options are described in detail in the Initial Settings View selection.

Node

The menu options displayed here are only available if the **Node…** check box is checked on the **General** tab.

Node options are described in detail in the Initial Settings View selection.

Drill Down tab options

To enable the options shown on the **Drill Down** tab control, make sure to check the **Drill Down** check box found in the Accessibility options located in the General View.

Drill Down

When a Drill Down is active (e.g., the user has double-clicked on the graph control), check this box to identify the graph control that you are opening (e.g., drilling down to), and any other controls that you need to hide and unhide (controls that are applicable to the new graph control can be unhidden, and controls that were applicable to the original graph control can be hidden).

Return

When the Drill Down tab control is active, check this box to identify the graph control that you are returning to from the drill down, and any other controls that you need to hide and unhide (controls that are applicable to the new graph control can be hidden, and controls that were applicable to the original graph control can be unhidden).

Tip

Use the **Drill Down** option in your initial, or parent graph control to target the drill down control. Use the **Return** option in the drill down, or child graph control to target the control to return to.

Controls to Hide/Unhide
Press the appropriate button to display update buttons to add, change, or remove, selected controls that will be automatically hidden or unhidden for the original graph control or the graph you are drilling down to. Select a Field Equate Label from the drop list.

Source Code support

The DrillDown feature of the Graph control is supported by a variety of properties and methods located in the GraphClass object.

For more information regarding the DrillDown method used to link graph controls, see the DrillDown method located in the ABC GraphClass.

Show tab options

The Show tab control completes your graph control design by designating optional Mouse, Diagram, and Node values and properties.

Select from the following options:

Mouse Sub-tab

The Mouse sub tab activates Mouse X and Y coordinates, and an optional label:

To show coordinates of the mouse as "Mouse:.."
The following options use a label with the X and Y coordinates:

on Tooltip
Check this box to display this information on the graph controls tool tip (Note: you need to have tool tips active in the *General* **View** settings.

on Window title
Check this box to display this information on the window title of the graph control.

on the control
Check this box to display this information on a target control that you select from the drop list.

on a zone of Status Bar
Check this box to display this information on the window's status bar in the zone designated by the spin control.

To show coordinates of the mouse:
The following options allow for finer display control of the X and Y coordinates:

on the control for coordinate X
Check this box to display the graph control's mouse X coordinate on a target control that you select from the drop list.

on the control for coordinate Y
Check this box to display the graph control's mouse Y coordinate on a target control that you select from the drop list.

on a zone of Status Bar for coordinate X
Check this box to display the graph control's mouse X coordinate on the window's status bar in the zone designated by the spin control.

on a zone of Status Bar for coordinate Y
Check this box to display the graph control's mouse X coordinate on the window's status bar in the zone designated by the spin control.

Diagram Sub-tab

The Diagram sub tab displays the name or value of the series point (node) region that the mouse has entered. This is a default value. You can modify this default name value by entering the appropriate source in the DiagramNameText method of the graph object. You can modify this default node value by entering the appropriate source in the DiagramName method of the graph object.

To show the name of diagram as "Diagram:…"
The following options use a label with the diagram name (the Name value designated in the Points option) in the following designated locations:

on Tooltip
Check this box to display the diagram information on the graph controls tool tip (Note: you need to have tool tips active in the *General* **View** settings.

on Window title
Check this box to display the diagram information on the window title of the graph control.

on the control
Check this box to display the diagram information on a target control that you select from the drop list.

on a zone of Status Bar
Check this box to display the diagram information on the window's status bar zone designated by the spin control.

To show the value of diagram
Diagram value refers to the text value of a target data point region that the mouse has moved in to.

on the control
Check this box to display the diagram value on a target control that you select from the drop list. Make sure that you select a control that displays a numeric data type

on a zone of Status Bar
Check this box to display the diagram value on the window's status bar zone designated by the spin control.

Node Name Sub-tab

The Node name refers to the **Field Name** specified in the **Points** tab control.

Note:

A node name and value will be displayed when the mouse is resting on top of the node area.

To show the name of node as "Node:…"
Example: "Node: Accounting" (*DEP:Name* is the value of Field Name)

on Tooltip
Check this box to display the node name information on the graph controls tool tip (Note: you need to have tool tips active in the *General* **View** settings.

on Window title
Check this box to display the node name information on the window title of the graph control.

on the control
Check this box to display the node name information on a target control that you select from the drop list.

on a zone of Status Bar
Check this box to display the node name information on the window's status bar in the zone designated by the spin control.

To show the name of node:
Example: "Accounting" (*DEP:Name* is the value of Field Name)

on the control
Check this box to display the node name information on a target control that you select from the drop list.

on a zone of Status Bar
Check this box to display the node name information on the window's status bar in the zone designated by the spin control.

Node Value Sub-tab

The Node Value is simply a calculated graph point expressed in (X ,Y) format

To show the value of node as "Node:

on Tooltip
Check this box to display the node value information on the graph controls tool tip (Note: you need to have tool tips active in the *General* **View** settings.

on Window title
Check this box to display the node value information on the window title of the graph control.

on the control
Check this box to display the node value information on a target control that you select from the drop list.

on a zone of Status Bar
Check this box to display the node value information on the window's status bar in the zone designated by the spin control.

To show the value of node:

on the control for coordinate X
Check this box to display the graph control's X node value on a target control that you select from the drop list.

on the control for coordinate Y
Check this box to display the graph control's Y node value on a target control that you select from the drop list.

on a zone of Status Bar for coordinate X
Check this box to display the graph control's X node value on the window's status bar in the zone designated by the spin control.

on a zone of Status Bar for coordinate Y
Check this box to display the graph control's Y node value on the window's status bar in the zone designated by the spin control.

Classes

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 in the **Global Properties** dialog. You may accept the default Application Builder Class specified in the **Global Properties** dialog (recommended), or you may specify your own or a third party class to override the default setting. 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.

GraphReport - Graph Control

The Graph Report Control template is the definitive tool that is designed to solve nearly all of your business printing graphing needs.

Due to the complexity and options needed for this template, a drop list divides the template prompts into four (4) distinct categories:

General *Initial values* *Data Sources* *Classes*

General

Refresh Automatically

Check this box to direct the templates to generate the appropriate graph refresh each time the window is refreshed, or in reports, each time a record is read. You can also control the refresh of the graph with the appropriate Graph Refresh Button or Code template.

Accessibility

Accessibility determines how much access or control you want to give to the user to control the graphing display. Not available in reports.

Initial Values and Data sources

These options are exactly the same as those found in the Graph Control used on window structures. Refer to that section for more information.

Class

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 in the **Global Properties** dialog. You may accept the default Application Builder Class specified in the **Global Properties** dialog (recommended), or you may specify your own or a third party class to override the default setting. 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.

GraphPrintButton - Print the Graph Control

The GraphPrintButton control template adds a button named *Print Graph*, and the underlying code necessary for printing a Graph control. Use this control template together with the Graph Control .

Edit the **Actions** only if you wish to add another, separate action to take place *after* printing. All the code necessary for managing the print job itself is handled automatically.

The **Actions** tab contains the following:

When Pressed
The standard set of prompts for buttons. Normally, when using a Control template, these prompts are not used.

GraphRefreshButton - Refresh the Graph Control

The GraphRefreshButton control template adds a button named *Refresh Graph*, and the underlying code necessary for refreshing a Graph control. Use this control template together with the Graph Control .

Edit the **Actions** only if you wish to add another, separate action to take place *after* refresh. All the code necessary for refreshing the graph control is handled automatically.

The **Actions** tab contains the following:

When Pressed
The standard set of prompts for buttons. Normally, when using a Control template, these prompts are not used.

GraphReturnButton - Return to the Graph Control

The GraphReturnButton control template adds a button named *Return to Graph*, and the underlying code necessary for returning to a target Graph control. Use this control template together with the Graph Control.

This template is useful when using a drill down configuration with multiple graph controls on the window. The drill down graph control is the control that is attached to the GraphReturnButton. When the graph is displayed, the *Return to Graph* button is displayed (using UNHIDE), and pressing the button returns to the parent, or target, graph control.

The **Actions** tab contains the following:

Control To Return From
Use the drop list to select the field equate that the button will return to. This should be the parent graph control, but is not limited to graph controls *only*.

When Pressed
The standard set of prompts for buttons. Normally, when using a Control template, these prompts are not used.

GraphRefreshCode - Refresh the Graph Control by Code

The GraphRefreshCode code template allows you to refresh the Graph Control contents at nearly any source embed point. This is useful when a data value is updated, and its value directly affects the graph control's contents.

The code template has no prompts, and simply calls the Graph object's Refresh method.

GraphRefreshCodeReport - Refresh the Graph Control by Code

The GraphRefreshCodeReport code template allows you to refresh the Graph Report Control's contents at nearly any source embed point (normally the TakeRecord method). This is useful when a data value is updated, and its value directly affects the graph report control's contents.

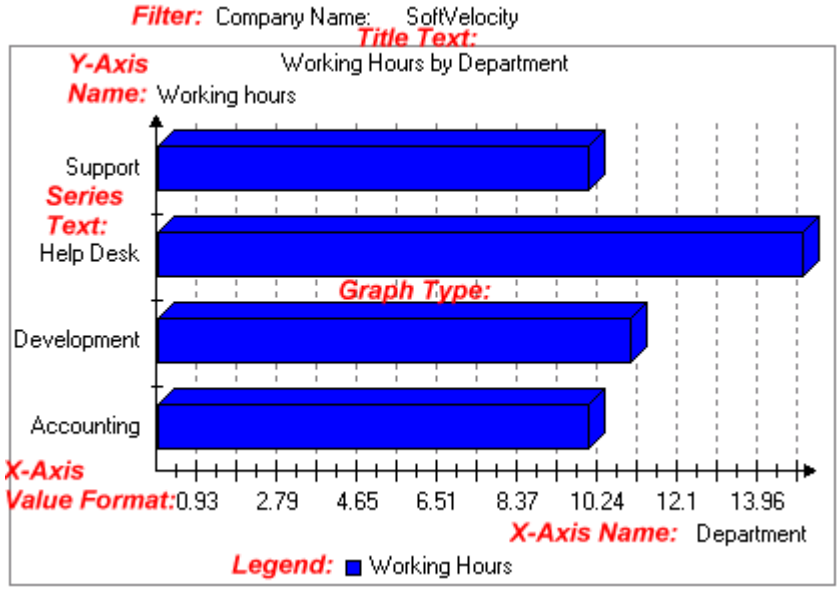
The code template has no prompts, and simply calls the Graph object's Refresh method.

Parts of a Graph

This help topic links basic graphing terminology with the template prompts used in the Clarion Graph control.

The Basics

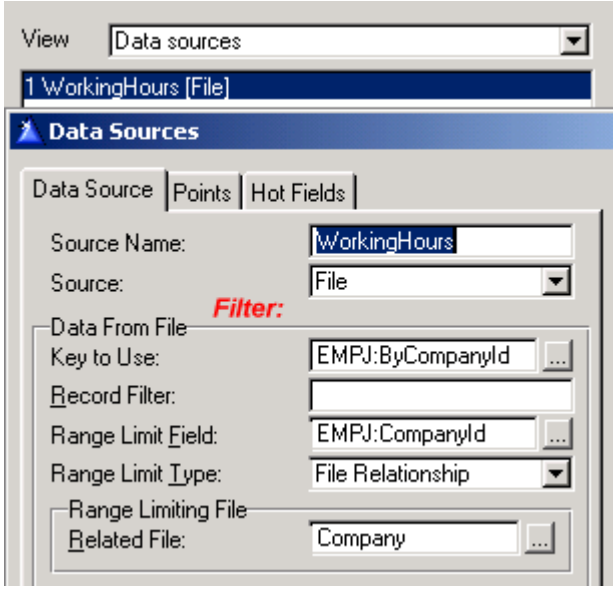
Consider the following graph:



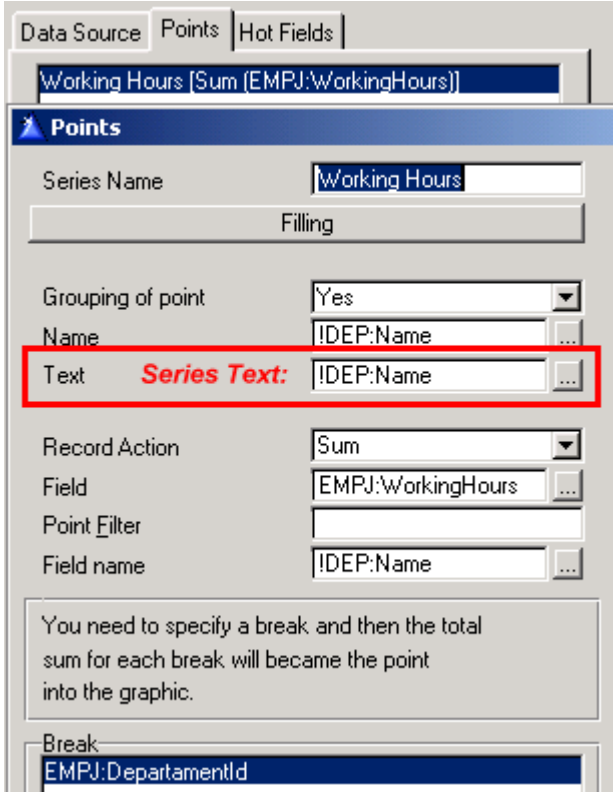
Items marked in **red** refer to labels not added by the graph control, but are used to define the graph's key elements.

What are we looking at?

The graph shown above plots a department's working hours devoted to a company. There is a single Data Source defined reading the *EmpJob* file and filtering the hours read by *CompanyId*.



In this graph, there is only one set of data points defined (called a **series**). This series draws the sum of working hours, and breaks when the DepartmentID (sp) changes:

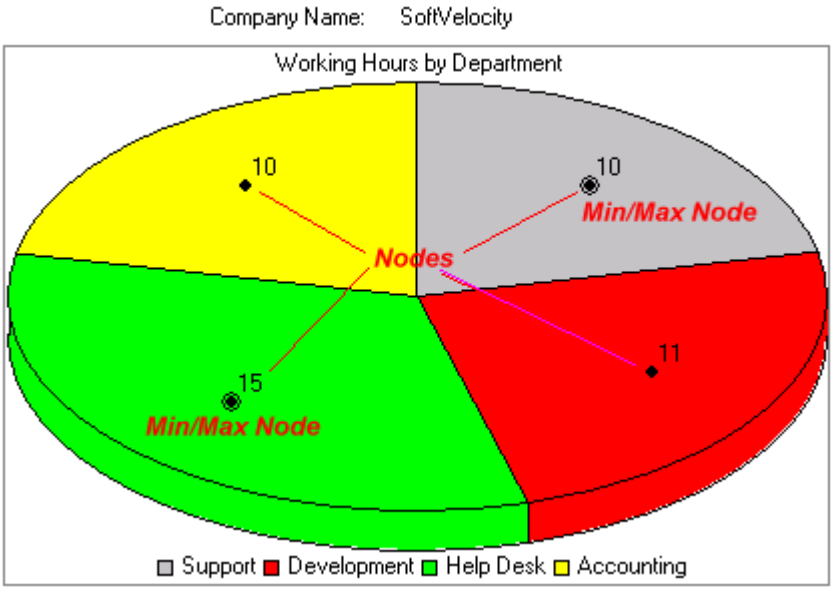


The other labels are Initial Values, and are initialized as follows:

Title Text	See *Initial Values* > *Title* > *Title Text*
Y-Axis Name	See *Initial Values* > *Axis* > *Y-Axis* > *Name*
X-Axis Name	See *Initial Values* > *Axis* > X-*Axis* > *Name*
Graph Type	See *Initial Values* > *Graph* > *GraphType*
X-Axis Value Format	See *Initial Values* > *Axis* > X-*Axis* > *Format of values* NOTE: If blank, this value uses default picture defined in dictionary.
Legend	See *Initial Values* > *Legend*> *Location*

Nodes

Examine the next graph shown below:



This pie graph is using the same data series points as the column graph above, but there are a few new graph parts that are now visible. The legend is now expanded to identify each piece of the pie chart, and the actual data values of each point in the data series is now displayed in a set of *nodes*.

The appearance of a node is dependant upon the graph type used. In addition, a node can be used to display additional information about the data point based on a number of template options.

Node information can be displayed in a window control, a tool tip, or on the window status bar.

Nodes are often used as "hot spots" for a drill down graph control. This allows a node value to be passed to the drill down graph control to provide fine control over the elements displayed.

The amount of node information that can be displayed is varied, and controlled by template options in many places.

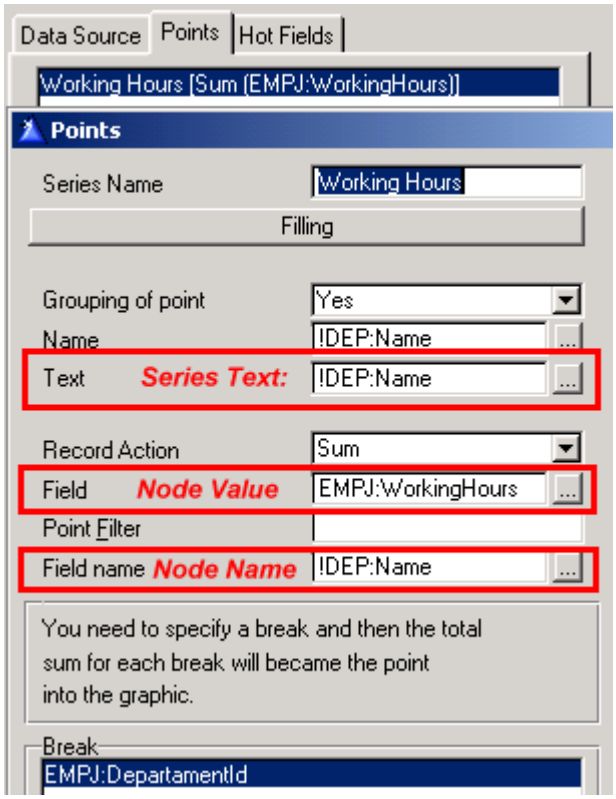
See:
View: *Initial Values* > *Node*

View: *Display options* > *PopUp Menu* > *Node*

View: *Display options* > *Show* > *Node Name*

View: *Display options* > *Show* > *Node Value*

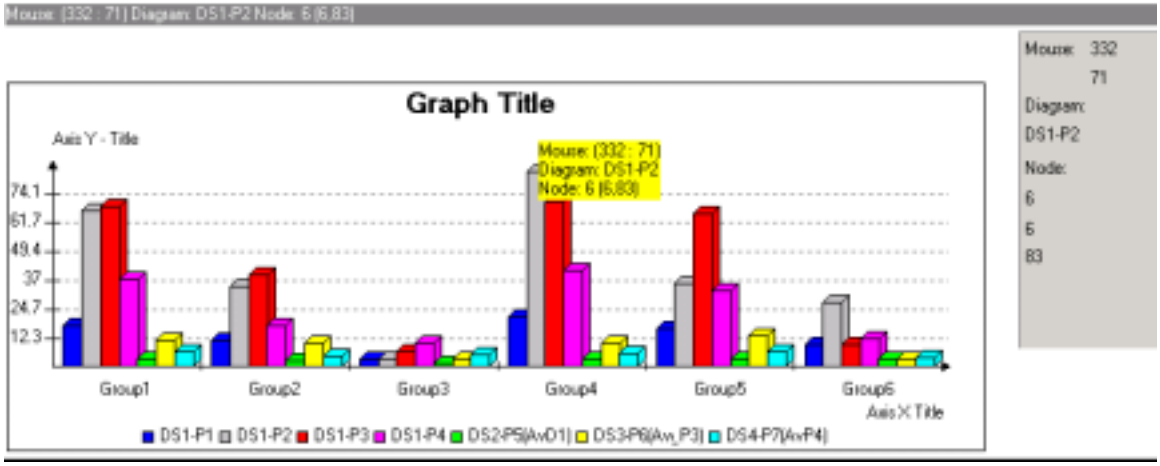
One last point concerning nodes; you have the option to show a node name or node value. These values are extracted from the *Points* dialog as shown below:



Advanced Graph Displays

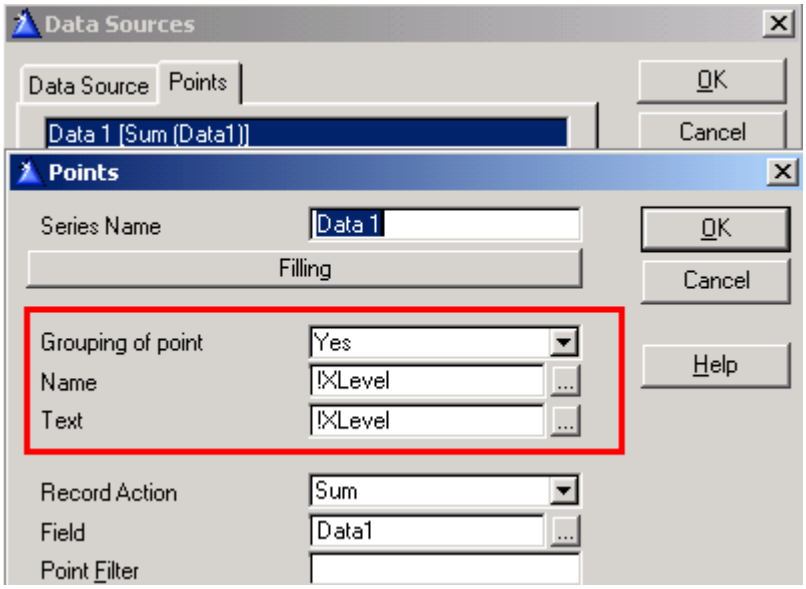
The most important rule to remember when designing a graph display is to *keep it simple*.

The following graph, captured from the shipping graph example, demonstrates how quickly a graph can become more complex:



This graph shows the "works". It is highly unlikely that you will ever need this level of display information, but you should only use this design for tutorial purposes.
The highlight of this graph display demonstrates multiple data points from multiple data sources. In reality, there is only one queue that is processed, but the last three data points rely on a specific average, and could process the queue in a different way. To summarize, one data source is used to process the first four data points (series), and each of the other data sources process an individual average series.

The seven data series are also grouped. This feature is controlled by the following settings:

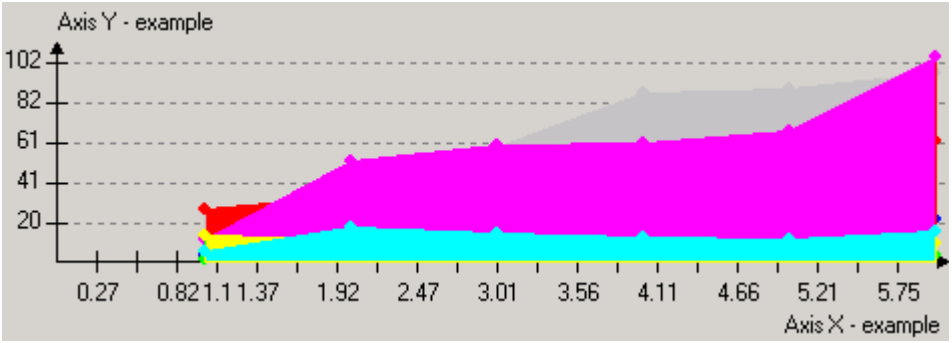


For each data series, **Grouping of point** is set to *Yes*, and the group is based on the value of *Xlevel* (in our example queue above, *XLevel* has six values stored – Group1, Group 2, etc.).

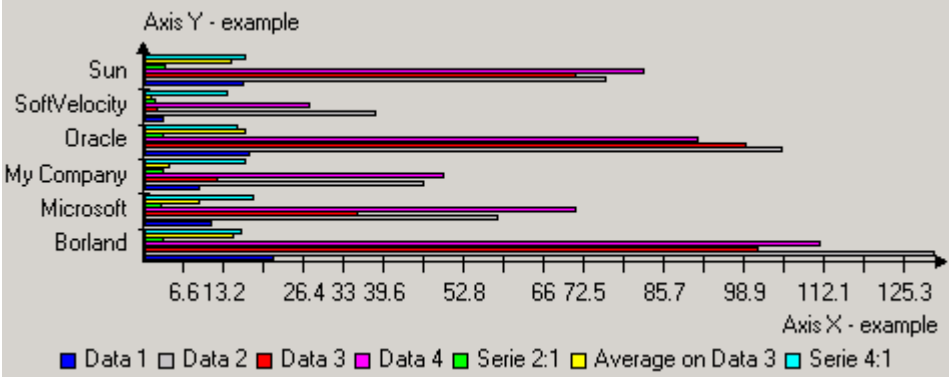
Again, care must be taken that your graph only displays information that is absolutely necessary.

Sample Graph Types

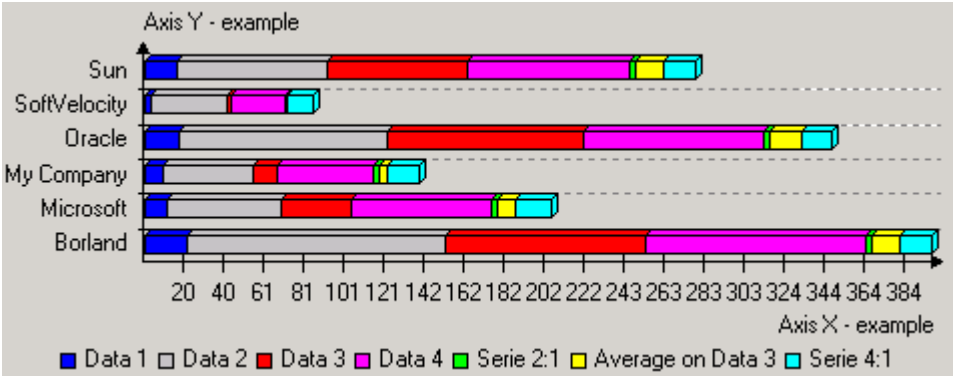
Area Graph



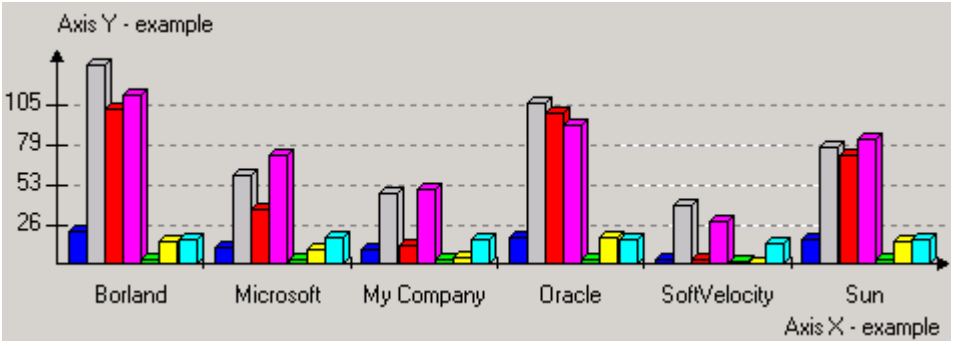
Bar Chart



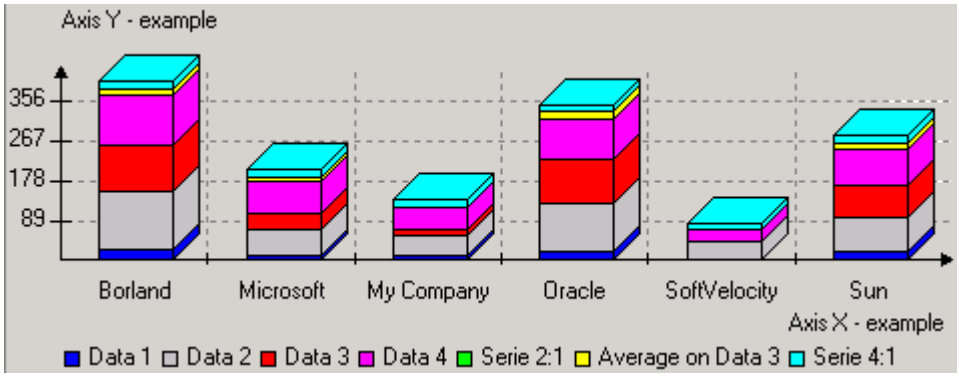
Bar Chart with Accumulators



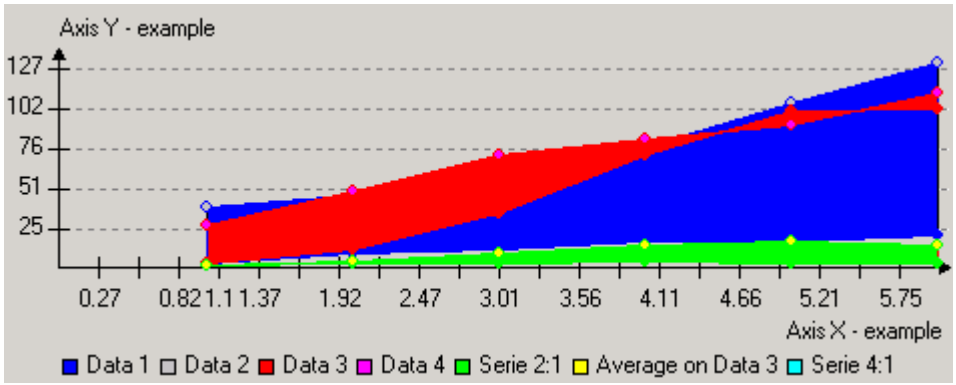
Column Chart



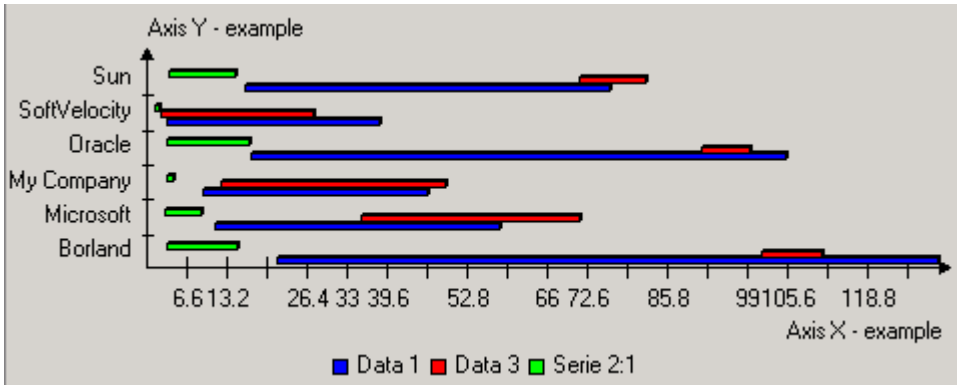
Column Chart with Accumulators



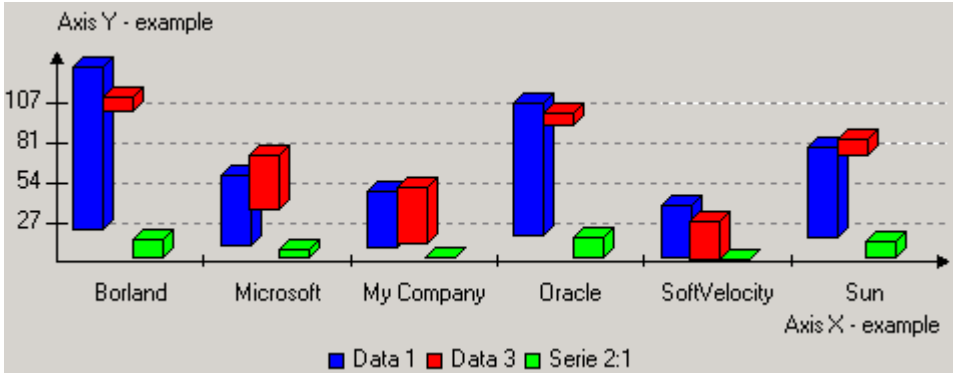
Floating Area



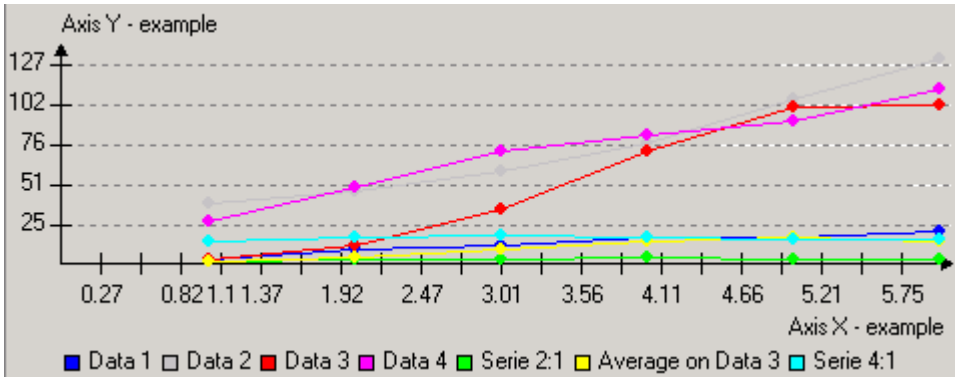
Floating Bar



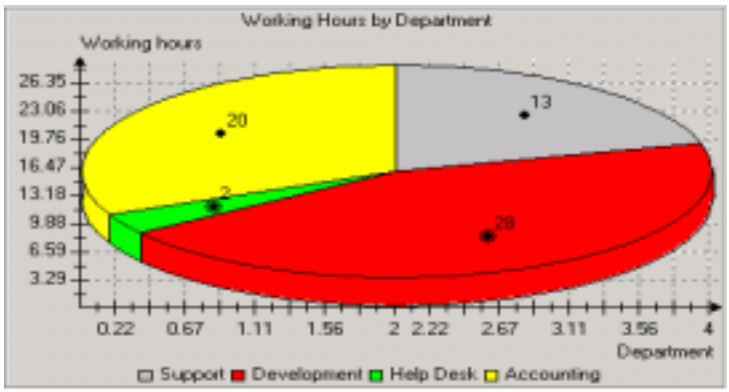
Floating Column



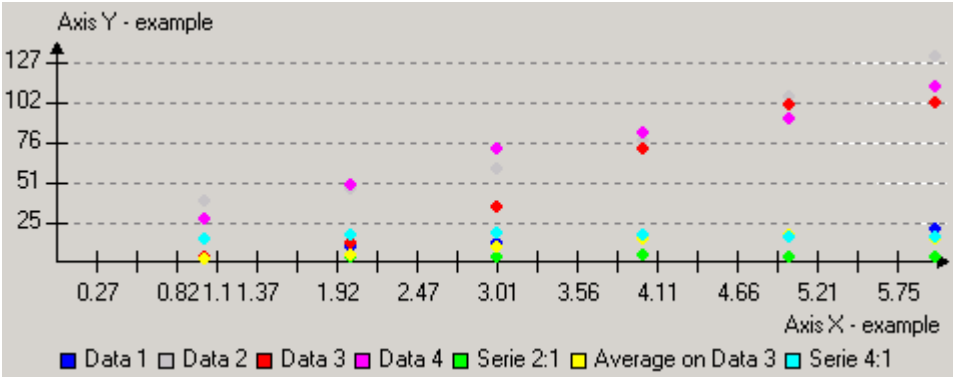
Line Graph



Pie Graph



Scatter Graph



Miscellaneous and Multi-Purpose Templates

This section covers the group of templates that cannot be grouped into a single category, and are normally used across many procedures.

Template types divide each subgroup. For more information on template types, please refer to the *Template Language Reference* documentation as well as the *Online User's Guide*.

Code Templates

CallABCMethod

The CallABCMethod template generates code to call an ABC Library object method. This template generates code similar to the following:

```
Default:City = INIMgr.Fetch('Preferences','City')
```

Object Name Select the label of the object from the list. The list contains all ABC compliant objects in scope for this procedure.

Method to Call Select the method to call from the drop-down list. Scroll the list horizontally or press the **Application Builder Class Viewer** button to see all the method parameters and return values.

Passed Parameters Type the parameter list to pass. Enclose the parameters in parentheses and separate them with commas. The parameters may be literal values, expressions, or variable names.

Return Value Assignment
Type the variable to receive the called method's return value. This field is only available for methods that return a value.

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.

Call a Procedure (Extended) code template

This code template is similar to the actions available in the Menu Editor, and adds extended features through the following template prompts:

Action Choose *Call a Procedure* or *Run a Program* from the drop list provided.

The remaining prompts are similar to those found in the Menu Editor for each menu item's actions, and are shown below:

Procedure Definition Prompts

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.

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.

Return Thread Variable
If the thread procedure called is prototyped to return a value, press the ellipsis button here to select a variable to receive the returned value.

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.

Return Value Variable
If the procedure called is prototyped to return a value, press the ellipsis button here to select a variable to receive the returned value.

Optionally check the **Reference Assign** check box if the variable defined is a reference variable.

Program Definition Prompts

Program Name Type the program name.

Parameters Optionally type a list of values that are passed to the program.

Call Procedure As Lookup code template

This Code template calls a procedure to select a record. It sets a variable called RequestCompleted to advise whether the lookup was successful.

Lookup Procedure Specifies the procedure to call.

Code before Type in any executable code to execute before performing the lookup. Multiple statements can be used if separated by a semicolon.

Code After, Completed Type in any executable code to execute after completing a lookup. Multiple statements can be used if separated by a semicolon.

Code After, Canceled Type in any executable code to execute if the lookup is canceled. Multiple statements can be used if separated by a semicolon.

Close Current Window code template

This code template simply posts an EVENT:CloseWindow, which tells the currently active window to close. There are no prompts to fill in.

DisplayPopupMenu

The DisplayPopupMenu template generates code to define and display a popup menu, and optionally, act on the end user's selection. You can set the popup menu items to mimic existing buttons on the window so that the associated menu item text matches the *button* text, is enabled only when the *button* is enabled, and, when selected, invokes the *button* action.

The DisplayPopupMenu template relies on the PopupClass to accomplish its tasks.

String variable for

Press the ellipsis (...) button to select or define a string variable to receive the end user's popup menu selection. After the popup menu displays, this variable contains the selected item's text minus any special characters. That is, the variable contains only characters 'A-Z', 'a-z', and '0-9'. If the resulting value is not unique for the menu, the PopupClass appends a sequence number to the value to make it unique.

You may interrogate this variable and perform actions depending on its value. If you rely on the PopupClass mimic capability to perform appropriate actions, then you can leave this field blank. See *Item Properties* for more information on mimic.

Build Menu From Choose how the popup menu and its items are defined. Choose from:

Menu String	Use the **Menu String** field to type the menu definition, then use the **Item Properties** to define each item's behavior.
Item List	Use the **Menu Items** button to define menu items one at a time.
INI File	Use the **Menu Description** field to name the INI file section which contains the menu definition. By default, the template code uses the global INIMgr object declared by the ABC Application template. If you have not specified an INI file to use, the INIMgr object uses Windows INI file. See *Template Overview—Global Options Tab*.

Menu Description Type the INI file section which contains the menu definition.

Menu String Type a menu definition string. The *Language Reference* describes the syntax for the menu definition string under the *selections* parameter for the POPUP command.

Item Properties Press this button to define the properties for each popup menu item. Only items specified in the Menu String are valid. You can set the popup menu items to mimic existing buttons on the window so that the associated menu item text matches the *button* text, is enabled only when the *button* is enabled, and, when selected, invokes the *button* action. You can also set the popup menu items to post an event to a control.

Menu Items Press this button to define the text for each popup menu item. You can set the popup menu items to mimic existing buttons on the window so that the associated menu item text matches the *button* text, is enabled only when the *button* is enabled, and, when selected, invokes the *button* action. You can also set the popup menu items to post an event to a control.

Classes Tab

Use the Classes tab to override the global settings for the Class. See Classes Tab.

GetFullDragSetting code template

This code template provides an easy way to detect the full window area dragging mode in your runtime application.

Target Variable Press the ellipsis button to select a variable name that will store the value of the FULLDRAG() API. This variable should be a numeric type.

Source Equivalence: ***Target Variable* = FULLDRAG()**

GetOSVersion code template

This code template provides an easy way to detect the operating system where your application is running by using the new PROP:WindowsVersion runtime property.

Target Variable Press the ellipsis button to select a variable name that will store the value of the runtime property. This variable should be a STRING or CSTRING type.

Source Equivalence: ***Target Variable* = SYSTEM{PROP:WindowsVersion}**

InitiateThread code template

When opening an MDI window from an Application Frame, you must initiate an execution thread. This Code template provides an easy way to initiate a thread.

When you START a procedure on its own thread, the procedure and its window operate independently of other threads in the same program; that is, the end user can switch focus between each execution thread at will. These are "**modeless**" windows.

If you don't initiate a new thread, the program behavior depends on whether the procedure's window has the MDI attribute. *A non-MDI* child window on the same thread as its parent, blocks access to all other threads in the program. This is an "**application modal**" window. When the application modal window closes, the other execution threads are available again. *An MDI* child window on the same thread as its parent, blocks access only to its parent window. When the MDI child window closes, its parent window regains focus.

In the **Prompts for Initiate Thread** dialog, simply name the procedure that opens the MDI window. Optionally, you can modify the size of the stack to allocate to the new execution thread. The default stack is 25,000 bytes.

You can optionally add a line of code to execute if the application was unable to open the thread. Type in the edit box labelled **Error Handling**. For example,

```
MESSAGE('Could not Start Thread','Error',ICON:HAND)
```

would display a message box with the halt (hand) icon, if the thread failed to start.

You can add a procedure name to call upon an error by typing the name of the procedure in the **Error Handling** box. You would then add the procedure to the **Application Tree** with the **Insert Procedure** command.

Lookup Non-Related Record code template

The LookupNonRelatedRecord template is used to perform a lookup of a value based on a relationship, whether it is or is not defined in the data dictionary (Ad hoc relation). You can add this Code template to the Lookup Up Related Records embed point.

Lookup Key Type in the key name or press the ellipsis (...) button to select the key from the File Schematic.

The lookup key is used to perform the lookup into the lookup file. This *must* be a unique key. If the key is a multi-component key, the other key elements must be primed before executing this Code template.

Lookup Field Type in the field name or press the ellipsis (...) button to select the field from the Component list.

The Lookup Field must be a component of the Lookup Key. This is the unique value within the lookup file.

Related Field Type in the related field or press the ellipsis (...) button to select it from the File Schematic.

The Related Field provides the unique value used to perform the lookup.

This template generates code similar to the following:

```
ST:StateCode = CUST:State                          ! Move value for lookup
Access:State.Fetch(ST:ByCode)                  ! Get value from file
```

SelectToolbarTarget code template

This Code template provides an easy way for developers to control which Browsebox in a given procedure is tied to the Toolbar navigation buttons (see *FrameBrowseControl*).

Toolbar Navigation Target

Select the Browsebox or the Form that is controlled by the FrameBrowseControl navigation buttons.

This Code template generates the following code:

```
DO BRW1::AssignButtons
```

where BRW1 identifies the Browsebox to navigate, or

```
DO FORM::AssignButtons
```

to tie the navigation buttons to the active Form procedure (that is, the SaveButton Control template).

ResizeSetStrategy

The ResizeSetStrategy template lets you override the default resize strategy for a particular control. It is designed exclusively for the **Set resize strategy** embed point for a specific control. See WindowResize extension template for more information on the default resize strategies.

Insert the code template at the **Set resize strategy** embed point for the control for which to set the resize strategy, then complete the following prompts.

Horizontal Resize Strategy

Specify how the control's width is determined when the end user resizes the window. Choose from:

Lock Width	The control's design time width does not change.
Constant Right Border	Locks right edge, moves left.

Vertical Resize Strategy

Specify how the control's height is determined when the end user resizes the window. Choose from:

Lock Height	The control's design time height does not change.
Constant Bottom Border	Locks bottom edge, moves top.

Horizontal Positional Strategy

Specify how the control's horizontal position is determined when the end user resizes the window. Choose from:

Lock Position	The control's left edge maintains a fixed distance (the design time distance) from parent's left edge.
Fix Right	The control's right edge maintains a proportional distance from parent's right edge.
Fix Left	The control's left edge maintains a proportional distance from parent's left edge.
Fix Center	The control's center maintains a proportional distance from parent's center.
Fix Nearest	Applies Fix Right or Fix Left, whichever is appropriate.

Vertical Positional Strategy

Specify how the control's vertical position is determined when the end user resizes the window. Choose from:

Lock Position	The control's top edge maintains a fixed distance (the design time distance) from parent's top edge.
Fix Bottom	The control's bottom edge maintains a proportional distance from parent's bottom edge.
Fix Top	The control's top edge maintains a proportional distance from parent's top edge.
Fix Center	The control's center maintains a proportional distance from parent's center.
Fix Nearest	Applies Fix Top or Fix Bottom, whichever is appropriate.

SetFullDragSetting code template

This code template provides an easy way to enable full window area dragging in your application.

Turn On Check this box to enable full window dragging in your application.

Use this code template to force full window dragging if you want to force this look and feel in your applications, or a user's operating system is having problems with the Enable Window Frame Dragging setting in the application's Global Properties.

The best place to include this template is any embed point that is available prior to opening the Application Frame.

Source Equivalence: **FullDrag(1)**

SetABCProperty

The SetABCProperty template generates code to set a public property of an ABC Library object. This template generates code similar to the following:

```
BRW2.ActiveInvisible = True
```

Object Name Select the label of the object from the list. The list contains all ABC compliant objects in scope for this procedure.

Property to Set Select the property to set from the drop-down list. See *ABC Library* for more information on these properties.

Value to Set Type a variable, constant, or valid Clarion expression to assign to the property.

Assign as Reference?
Check this box to generate a reference assignment (*object.property &= value*). Clear the box to generate a simple assignment (*object.property = value*). See *Reference Assignments* in the *Language Reference* for more information.

SetProperty code template

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

Control Select the field equate label for one of the window controls from the drop down list.

Property Select the runtime property to set from the drop down list.

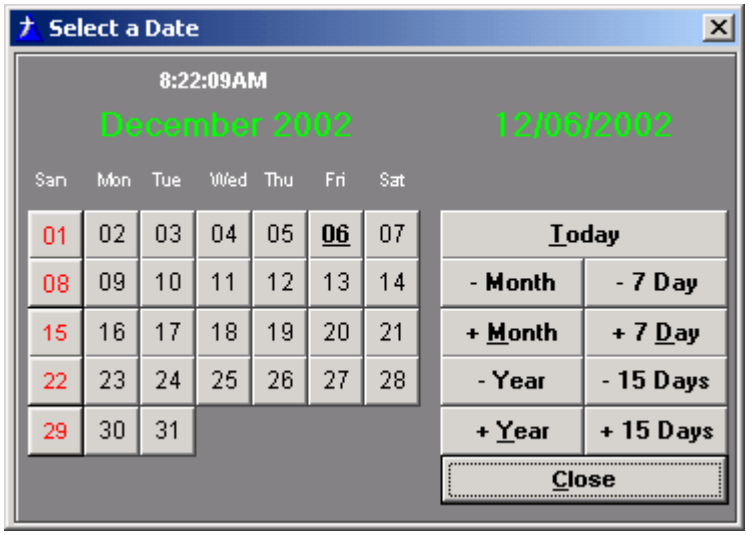
Value The label of a variable, a constant, or an expression to assign to the selected runtime property.

This Code template generates the following code:

```
?MyControl{PROP:Whatever} = value
```

Control Templates

Calendar Button Control



The Calendar Button Control Template is a button control that, when pressed, displays a full-feature calendar. A date value is returned to a targeted entry control. There are simple options to control the calendar title, extent of display features, and refresh window options.

The CalendarButton template provides the following prompts:

Control

Select the associated ENTRY control for which to send the date value you selected on the calendar control.

Calendar Title

Enter a string value that you wish to use as the title of your calendar control.

Refresh Window

Check this box to force a refresh of the window controls when returning to the calling procedure.

Calendar Control – Classes Tab

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 in the **Global Properties** dialog. You may accept the default Application Builder Class specified in the **Global Properties** dialog (recommended), or you may specify your own or a third party class to override the default setting. 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.

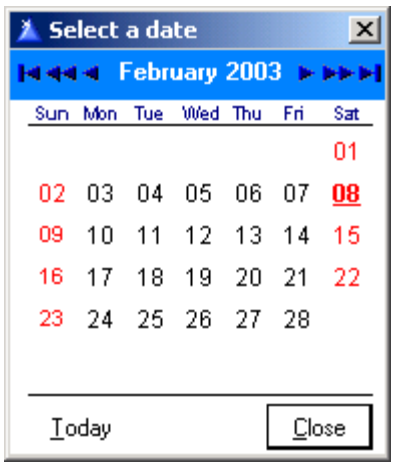
Click here to view the Global Calendar Configuration dialog.

Clarion provides you with three types of Calendar Classes that you may use with the CalendarButton control template, or, in hand coded projects if needed.

Select from the following classes:

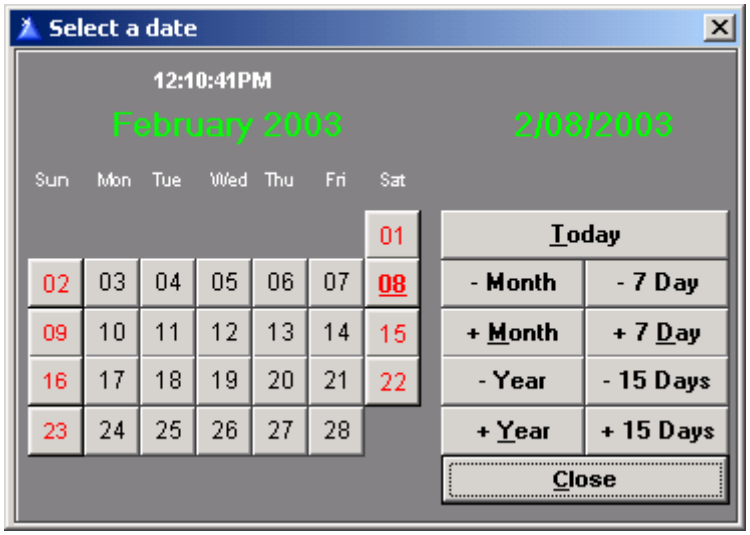
CalendarClass

The CalendarClass displays a simple and functional calendar that was designed to be compatible with the Windows XP look.



CalendarSmallClass

The CalendarSmallClass displays a calendar that is more compatible with other non-XP style environments:



CalendarBaseClass

The default source code for the CalendarBaseClass is located in the ABUTILUI.CLW file (located in the LIBSRC directory). You must include the default window and corresponding source code in the respective DATA and CODE sections of the CalendarBaseClass derived ASK method.

Cancel Button control template

The CancelButton template adds a single button control marked **Cancel**. This button lets the user close a window and it provides a convenient place for the developer to add code to "undo" before closing down the procedure. The generated source code sets a "Request Cancelled" flag and closes down the window procedure.

The CancelButton template provides no configuration options.

You can insert the executable code you need to "clean up" at an embed point.

When Pressed The standard set of prompts for buttons Normally, when using a Control template, these prompts are not used.

Close Button control template

The CloseButton template adds a single button control marked **Close**. The generated source code sets a "RequestCancelled" flag and closes down the window procedure.

The CloseButton template provides no configuration options.

The **Action** tab provides the following prompts:

When Pressed The standard set of prompts for buttons Normally, when using a Control template, these prompts are not used.

DOSFileLookup template

The DOSFileLookup template adds an ellipsis (...) button which opens the standard Windows file dialog.

You can specify the file masks, the default directory and filename, and the variable to receive the filename selected by the end user.

In addition, you may optionally allow the selection of multiple files and specify the code to process each selected file. The template generates a LOOP to process all the selected files.

The DOSFileLookup template provides the following prompts:

General

File Dialog Header
Type the text for the caption of the Windows file dialog.

DOS Filename Variable
Press the ellipsis (...) button to choose a variable to receive the end user's choice from the **File Schematic** dialog. You can also type the variable name directly into the entry box.

Default Directory
Specify the starting directory for the Windows file dialog. If blank, the file dialog opens to the working directory.

Default Filename
Specify the initial filename for the Windows file dialog. If blank, the file dialog opens with no initial filename.

Return to original directory when done
Check this box to reset the working directory to its value prior to the file lookup.

Long Filenames?
Check this box to enable the Windows file dialog to allow the selection of files with long filenames.

Multi-Select?
Check this box to allow selection of one *or more* files.

Action For Each Selection

Type a valid Clarion language statement to execute for each selected file--typically a procedure call. You may want to pass the **FileName Variable** as a parameter to the procedure.

The template generates a LOOP to execute the code you specify for each selected file. The generated code reloads the **FileName Variable** with the appropriate filename for each loop cycle.

Assign to Image Control

Check this box if you wish to assign the selected filename's contents to an image control. Use the **Image Control** drop list to select the image control's Field Equate Label.

Resize to control size

Check this box to designate that the filename's contents assigned to the target Image control will be resized to fit the control.

Set Default Mask Value

Press this button to set the default file masks to use when selecting a valid image file to load into the target image control.

File Masks

Use a variable file mask

Check this box to supply the file mask with a variable. This enables the **Variable Mask Value** field to name the variable, and disables the **Mask Description, File Mask,** and **More File Masks** prompts.

Mask Variable

Names the variable that contains the file mask. See *FILEDIALOG* in the *Language Reference* for information on the contents of this variable.

File Mask Description

Type a file type description. The string appears in the drop-down list in the Windows file dialog. You can add additional masks by pressing the **More File Masks** button.

File Mask

Type a file mask specification, such as "*.TXT" or use multiple patterns for this mask separating each with a semicolon, such as "*.BMP;*.GIF". You can add additional masks by pressing the **More File Masks** button.

More File Masks

Press this button to add additional file masks. These masks are available to end the user through the **List files of type** drop-down list in the Windows file dialog.

Window Update Options

Update entire window?

Check this box to refresh the contents of all window controls after the file selection and processing is complete. Clear the box to select specific fields to refresh.

Update Selected Fields

Press this button to select specific fields to refresh after the file selection and processing is complete. The template generates a DISPLAY statement for each field you specify. See *DISPLAY* in the *Language Reference.*

Classes Tab

Use the Classes tab to override the global settings for the Class. See Classes Tab.

DynamicImage

The DynamicImage control provides an image control that must be placed within an existing BrowseGrid control. This template provides the BrowseGrid with the correct image associated with the column specified by the Field to drive image prompt. The image is a clickable image that can perform an action associated with any control on the window. The DynamicImage control requires a BrowseGrid control.

The DynamicImage template provides the following prompts:

Field to drive image Specifies the column from the table that designates the image file name.

Classes

The classes tab lets you control the class (and object) the template uses. You may accept the default Application Builder Class and it's object (recommended) or you may specify your own or a third party class. 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.

Field Lookup Button control template

The FieldLookupButton template provides an ellipsis (...) button that lets you "look up" the value from a lookup file, such as a state file. CLICK next to an input control to place the lookup button.

The FieldLookupButton template provides the following prompts:

Control with Lookup

Select the associated control for which to perform the lookup by choosing its field equate label from the drop-down list. Typically this is an ENTRY control.

The selected control must have an associated lookup procedure. To provide the lookup procedure, RIGHT-CLICK on the control, then choose **Actions** to access its prompts.

FileDrop Template

The FileDrop template places a file-loaded scrollable drop-down list on a window. At runtime, the end user can select an item from the list, then assign a value from the selected item's record to a specified target field. You may display one field (such as a description field) but assign another field (such as a code field) from the selected record (see Using drop-down lists to Lookup Records).

Tip: Set the DROP attribute to zero (0) to display a list box rather than a drop-down list.

Immediately before you place the FileDrop Control template on your window, the Application Generator prompts you to specify the file to display in the drop-down list. Specify the file in the **Select Field** dialog. You will also need to select a field to serve as the USE variable for the LIST; however, the field you select is only significant if you are displaying one field but assigning another).

Immediately after you place the FileDrop Control template, the Application Generator opens the List Box Formatter so you can specify the fields to display in your list. You may specify the field containing the lookup value as well as other fields from the same or related files.

After you specify your list fields and return to the window under construction, right-click the control, then choose **Actions** from the popup menu to complete the following FileDrop options:

General

Field to Fill From

The field in the lookup file whose value is assigned to the Target Field. Press the ellipsis (...) button to select from the **Select Field** dialog.

Target Field

The field that receives the value from the Field to Fill From. Press the ellipsis (...) button to select from the **Select Field** dialog.

More Field Assignments

Press this button to specify additional value assignments from the selected item's record.

Record Filter

Type a valid Clarion expression to limit the contents of the list to only those records causing the expression to evaluate to true (nonzero or non-blank). The procedure loops through all displayable records to select only those that meet the filter. Filters are generally much slower than Range Limits.

You must BIND any file field, variable, or EQUATE that is used in a filter expression. The **Hot Fields** tab lets you BIND fields.

Default to first entry if USE variable empty

Check this box to provide an initial default selection--the drop-down list is never initially empty (unless the first file record is a blank one).

Range Limits

This tab is only available if you specify a Key for the File in the **Table Schematic Definition** dialog. Because range limits use keys, they are generally much faster than filters.

Range Limit Field

In conjunction with the **Range Limit Type**, specifies a record or group of records for inclusion in the process. Choose a key field on which to limit the records by pressing the ellipsis (...) button.

Range Limit Type

Specifies the type of range limit to apply. Choose one of the following from the drop-down list.

Current Value

Limits the key field to its current value.

Single Value

Lets you limit the key field to a single value. Specify the variable containing that value in the **Range Limit Value** box.

Range of Values

Lets you limit the key field to a range of values. Specify the variables containing the upper and lower limits of the range in the **Low Limit Value** and **High Limit Value** boxes.

File Relationship

Lets you limit the key field to the current value in a related (parent) file. Press the **Related file** ellipsis (...) button to choose the range limiting file. This limits the process to include only those child records matching the current record in the parent file. For example, if your report was a list of Orders, you could limit the process to only those orders for the current Customer.

Colors

This tab is only available if you check the **Color Cells** box in the List Box Formatter. It displays a list of the FileDrop columns which may be colored.

To specify the default colors and any conditional colors, highlight the column's field name, then press the **Properties** button. This opens the **Customize Colors** dialog.

Customize Colors

This dialog lets you specify the default and conditional Foreground and Background colors for normal (unselected) rows; and for selected rows.

Conditional Color Assignments

Below the default colors section is the **Conditional Color Assignments** list. This list lets you set colors to apply when an expression evaluates to true (nonzero or non-blank). To add an expression and its associated colors, press the **Insert** button.

At run-time the expressions are evaluated, and the colors for the first true expression are used.

Icons

This tab is only available if you check the **Icons** box in the List Box Formatter. It displays a list of the FileDrop columns which can display icons.

To specify default icons and any conditional icons, highlight the column's field name then press the **Properties** button. This opens the **Customize Icons** dialog.

Customize Icons

This dialog lets you specify the default icon and conditional icons for the FileDrop column.

Default Icon

The default icon to display. Type the icon (.ICO) filename.

Conditional Icon Usage

Below the **Default Icon** section is the **Conditional Icon Usage** list. This list lets you set icons to apply when an expression evaluates to true (nonzero or non-blank). To add an expression and its associated icon, press the **Insert** button.

At run-time the expressions are evaluated, and the colors for the first true expression are used.

Styles

This tab is only available if you check the **Style** box in the List Box Formatter. It displays a list of the List Box columns that may have applied styles.

Tip

To specify the default styles and any conditional styles, highlight the column's field name, then press the Properties button. This opens the Customize List Box Styles dialog. A default style may also be defined on the List Box Formatter Appearance tab.

Create GreenBar Effect

Check this box to create a GreenBar effect (alternating colors on each row) on your list box. You will be prompted to select two styles to use that represent the appearance of each alternating row

Default Style

This entry lets you specify the default style to be used for the column.

Conditional Styles

This list lets you define the styles to apply when an expression evaluates to true (nonzero or non-blank). To add an expression and its associated colors, press the Insert button.

Condition
Provide a valid Clarion expression that when evaluates to true (nonzero or non-blank) will cause the Style to be applied.

Style
Define the style number that will be applied to the column when the Condition evaluates to true (nonzero or non-blank).

At run-time the expressions are evaluated, and the styles for the first true expression are used.

All Styles must be defined. For more information on defining styles see FORMAT (set LIST or Combo layout).

Tooltips

This tab is only active if you check the **Tooltip** box in the List Box Formatter. It displays a list of the List Box columns which may have applied tool tips. Press the **Properties** button to display the *Customize BrowseBox Tooltips* dialog.

Tooltip variable

You can specify a default tool tip string value in the List Box Formatter. This entry lets you override the default tooltip to use a value contained in a variable. Press the ellipsis button to select a variable that will contain the text of your column's tool tip.

Hot Fields

When you select the Hot Fields tab, you can specify fields not populated in the list to add to the QUEUE. When scrolling through the file, the generated source code reads the data for these fields from the QUEUE, rather than from the disk. This speeds up list box updates.

Specifying "Hot" fields also lets you place controls outside the FileDrop that are updated whenever a different record is selected in the list box. Elements of the Primary Key and the current key are always included in the QUEUE, so they do not need to be inserted in the Hot Field list.

This dialog also lets you BIND a field. You must BIND any field, variable, or EQUATE that is used in a filter expression or as a field to total. If the field you are BINDing is not in the VIEW, check the **Not in View** checkbox.

Sort Fields

This tab lets you add fields by which the items in the drop-down list are sorted. The sort fields are in addition to any Key specified for the FileDropCombo. Press the **Insert** button to add fields to the list.

Classes Tab

Use the Classes tab to override the global settings for the Class. See Classes Tab.

Other Prompts

The List Properties for this control are the same as for a list; however, the following prompts may require some additional explanation:

Use	Takes either a field equate label, or the label of a variable to receive the value from the first field populated in the list. In the FileDrop Control template context, this functionality is replaced by the more flexible Target Field setting.
From	This field defaults to Queue:FileDrop. Queue:FileDrop is the label of the QUEUE the template uses to fill the list. Typically, you should not change this value.
Mark	Takes the label of the Queue:FileDrop:Mark QUEUE field to allow the user to select more than one item from the list. The Queue:FileDrop:Mark field contains 1 for selected items and 0 for unselected items.

File Drop Combo control template

The FileDropCombo template generates code to display a data file in a scrollable list, select one of the records from the list, then assign a value from the selected record to a specified target field. Note that you may display one field (such as a description field) but assign another field (such as a code field) from the selected record (see Using drop-down lists to Lookup Records). Also, because the template is based on a COMBO control, the generated code accepts entry values that may not exist in the displayed list and optionally adds these new values to the lookup file.

Immediately before you place the FileDropCombo Control template on your window, the Application Generator prompts you to specify the file to display in the drop-down list. Specify the file in the **Select Field** dialog. You will also need to select a field from the file to serve as the USE variable for the COMBO. The USE variable is significant when you Allow Updates from the FileDropCombo or when you display one field but assign another.

Immediately after you place the FileDropCombo Control template, the Application Generator opens the List Box Formatter so you can specify the fields to display in your list. You may specify the field containing the lookup value as well as other fields with associated information.

After you specify your list fields and return to the window under construction, RIGHT-CLICK the control, then choose **Actions** from the popup menu to complete the following FileDropCombo options:

General

Field to Fill From

The field in the lookup file whose value is assigned to the Target Field. Press the ellipsis (...) button to select from the **Select Field** dialog.

Target Field

The field that receives the value from the Field to Fill From. Press the ellipsis (...) button to select from the **Select Field** dialog.

More Field Assignments

Press this button to specify additional value assignments from the selected item's record.

Record Filter

Type a valid Clarion expression to limit the contents of the list to only those records causing the expression to evaluate to true (nonzero or non-blank). The procedure loops through all displayable records to select only those that meet the filter. Filters are generally much slower than Range Limits.

You must BIND any file field, variable, or EQUATE that is used in a filter expression. The **Hot Fields** tab lets you BIND fields.

Default to first entry if USE variable empty

Check this box to provide an initial default selection--the drop-down list is never initially empty (unless the first file record is a blank one).

Remove duplicate entries

Check this box to remove duplicates from the list.

Keep View synchronized with Selection?

Check this box to update the VIEWs record buffers to match the selected item.

Automatic Entry Field Completion

Check this box to allow the entry control to act as a incremental locator. As you enter characters, the template locates and populates the closest matching record in the list box (equal to or greater than the current entry field's contents).

Case Sensitive matches?

Check this box to consider case when matching entered values with values in the lookup file.

Range Limits

This tab is only available if you specify a Key for the File in the **Table Schematic Definition** dialog. Because range limits use keys, they are generally much faster than filters.

Range Limit Field

In conjunction with the **Range Limit Type**, specifies a record or group of records for inclusion in the process. Choose a key field on which to limit the records by pressing the ellipsis (...) button.

Range Limit Type

Specifies the type of range limit to apply. Choose one of the following from the drop-down list.

Current Value

Limits the key field to its current value.

Single Value

Lets you limit the key field to a single value. Specify the variable containing that value in the **Range Limit Value** box.

Range of Values

Lets you limit the key field to a range of values. Specify the variables containing the upper and lower limits of the range in the **Low Limit Value** and **High Limit Value** boxes.

File Relationship

Lets you limit the key field to the current value in a related (parent) file. Press the **Related file** ellipsis (...) button to choose the range limiting file. This limits the process to include only those child records matching the current record in the parent file. For example, if your report was a list of Orders, you could limit the process to only those orders for the current Customer.

Colors

This tab is only available if you check the **Color Cells** box in the List Box Formatter. It displays a list of the FileDropCombo columns which may be colored.

To specify the default colors and any conditional colors, highlight the column's field name, then press the **Properties** button. This opens the **Customize Colors** dialog.

Customize Colors

This dialog lets you specify the default and conditional Foreground and Background colors for normal (unselected) rows; and for selected rows.

Conditional Color Assignments

Below the default colors section is the **Conditional Color Assignments** list. This list lets you set colors to apply when an expression evaluates to true (nonzero or non-blank). To add an expression and its associated colors, press the **Insert** button.

At run-time the expressions are evaluated, and the colors for the first true expression are used.

Icons

This tab is only available if you check the **Icons** box in the List Box Formatter. It displays a list of the FileDropCombo columns which can display icons.

To specify default icons and any conditional icons, highlight the column's field name then press the **Properties** button. This opens the **Customize Icons** dialog.

Customize Icons

This dialog lets you specify the default icon and conditional icons for the FileDropCombo column.

Default Icon

The default icon to display. Type the icon (.ICO) filename.

Conditional Icon Usage

Below the **Default Icon** section is the **Conditional Icon Usage** list. This list lets you set icons to apply when an expression evaluates to true (nonzero or non-blank). To add an expression and its associated icon, press the **Insert** button.

At run-time the expressions are evaluated, and the colors for the first true expression are used.

Styles

This tab is only available if you check the **Style** box in the List Box Formatter. It displays a list of the List Box columns that may have applied styles.

Tip

To specify the default styles and any conditional styles, highlight the column's field name, then press the Properties button. This opens the Customize List Box Styles dialog. A default style may also be defined on the List Box Formatter Appearance tab.

Create GreenBar Effect

Check this box to create a GreenBar effect (alternating colors on each row) on your list box. You will be prompted to select two styles to use that represent the appearance of each alternating row

Default Style

This entry lets you specify the default style to be used for the column.

Conditional Styles

This list lets you define the styles to apply when an expression evaluates to true (nonzero or non-blank). To add an expression and its associated colors, press the Insert button.

Condition
Provide a valid Clarion expression that when evaluates to true (nonzero or non-blank) will cause the Style to be applied.

Style
Define the style number that will be applied to the column when the Condition evaluates to true (nonzero or non-blank).

At run-time the expressions are evaluated, and the styles for the first true expression are used.

All Styles must be defined. For more information on defining styles see FORMAT (set LIST or Combo layout).

Tooltips

This tab is only active if you check the **Tooltip** box in the List Box Formatter. It displays a list of the List Box columns which may have applied tool tips. Press the **Properties** button to display the *Customize BrowseBox Tooltips* dialog.

Tooltip variable

You can specify a default tool tip string value in the List Box Formatter. This entry lets you override the default tooltip to use a value contained in a variable. Press the ellipsis button to select a variable that will contain the text of your column's tool tip.

Update Behavior

This tab lets you use the entry portion of the COMBO to initiate adding a new record to the lookup file. If the user types a value in the entry box that is not already in the list, the generated code can add a new record directly, or it can call a separate procedure to add the new entry.

Allow Updates Clear this box to accept entries that do not exist in the lookup file. The new (unvalidated) entries are *not* added to the lookup file.

Check this box to add new entries to the lookup file, and to enable the **Update Procedure** prompt.

Update Procedure Name the procedure to call to add the new record, or leave this field blank if no update procedure is needed.

No update procedure is needed for lookup files with only one required field (the field specified by the COMBO's USE variable). Non-USE fields are CLEARed, unless range limited or auto-incremented.

Hot Fields

Use the Hot Fields tab to specify fields to add to the QUEUE that are not displayed in the list. When scrolling through the file, the generated source code reads the data for these fields from the QUEUE, rather than from the disk. This speeds up list box updates.

Specifying Hot Fields effectively lets you update other controls whenever a new record is selected in the list box. Elements of the Primary Key and the current key are always included in the QUEUE, so they do not need to be inserted in the Hot Field list.

Press the **Insert** button to add fields to the list.

If the field you are BINDing is not in the VIEW, check the **Not in View** checkbox.

Sort Fields

This tab lets you add fields by which the items in the drop-down list are sorted. The sort fields are in addition to any Key specified for the FileDropCombo. Press the **Insert** button to add fields to the list.

Classes Tab

Use the Classes tab to override the global settings for the Class. See Classes Tab.

Other Prompts

The List Properties for this control are the same as for a list; however, the following prompts may require some additional explanation:

Use Takes either a field equate label or the label of a variable to receive the value from the first field populated in the list. In the FileDropCombo Control template context, the assignment functionality is replaced by the more flexible Target Field; however, the USE variable is significant when you Allow Updates from the FileDropCombo (see *Update Behavior* for more information).

From This field defaults to Queue:FileDropCombo. Queue:FileDropCombo is the field equate label of the QUEUE the template generates to fill the list. Typically, you should not change this value.

Mark Takes the label of the Queue:FileDropCombo:Mark QUEUE field to allow the user to select more than one item from the list. The Queue:FileDropCombo:Mark field contains 1 for selected items and 0 for unselected items.

Help Button control template

The Help Button control template populates a help button on a window that calls the program's help file when pressed. The help topic default is the Help ID that you have defined for the window.

There are no special prompts or actions for this control template.

OLE control template

This control template helps you implement OLE and OCX controls using the Clarion OLE container control.

For OLE controls the template generates code to activate and deactivate the OLE server and to assign the OLE object to a BLOB for saving.

For OCX controls the template generates appropriate MAP include statements to support the OCX, "shell" callback procedures for the OCX, and code to register the callback procedures. Depending on the OCX used, you may need to add additional functionality to the callback procedures.

OLE Properties

The OLE control template provides the following prompts:

Callback Generation Callback functions are called by Clarion's runtime library (they are not called directly by your program) when it needs to pass information to your program concerning the OCX. Your callback functions should quickly process the information and respond to the caller. The OLE template can generate shell callback functions and their prototypes so you can concentrate on any special functionality required by the callback.

Event Handler Check this box to generate an event handling callback function. Clarion's runtime library calls this function when an OCX event needs to be passed to your program.

Property Change Check this box to generate a property change callback function. Clarion's runtime library calls this function when an OCX property is changed.

Property Edit Check this box to generate a property edit callback function. Clarion's runtime library calls this function when an OCX property change is requested, but permission to make the change is required from your program before making the change.

Include OCX.CLW in global MAP
Check this box to include prototypes for OCX related functions (callbacks, register callbacks, unregister callbacks, etc.) in your program's global MAP.

Include OCXEVENT.CLW in global data section
Check this box to include common OCX event equates in your program's global data section.

OLE BLOB field Specify a BLOB to contain the OLE object, or press the ellipsis (...) button to select the BLOB from the **Select Field** dialog.

Extension Templates

This section describes the user interface that is used with the default Clarion Extension templates. See the *Template Language Reference* PDF for more information regarding Extension templates.

Additional Sort Fields Dialog

The Additional Sort Fields Dialog is a built-in extension of the Browse, Process, and Report procedures. Its function is to add additional sort fields to the default sequence that data is processed. This sequence can be applied to the procedure's default sort behavior, or to conditional behavior if supported. If no default sort is specified (no key), the additional sort fields dialog sets the primary sort.

This dialog is supported in both ABC and Clarion template chains. The sort sequence selected applies to the appropriate ABC method or ORDER property required.

Sort Type

Manual
Use the ellipsis to select a field name from the *Select Column* dialog. This allows you to manually construct a sort string using the following syntax:
*"+FieldLabelOne,-FieldLabelTwo,…***"**
An incorrect string will result in no sorting, and may generate a compile error.

Assisted
The Assisted method is designed to build a correct sort string. Use the update buttons provided to add your sort fields to the list. In the additional dialog, provide the following information:

Field
Use the ellipsis to select a field name from the *Select Column* dialog.

Order

Select Ascending or Descending from the drop list, to designate an ascending or descending sort order.

ConditionalAssisted
Valid for Report and Process only. The Conditional Assisted sort type adds a condition to the assisted method, where the additional sort fields will only be applied when the condition is true.

Condition
Use the ellipsis to select a field name from the *Select Column* dialog. When this field or column is non-zero, the additional sort fields will be applied.

Field
Use the ellipsis to select a field name from the *Select Column* dialog.

Order
Select Ascending or Descending from the drop list, to designate an ascending or descending sort order.

DynamicSorting
Valid for Report and Process only. This Sort Type generates a dynamic sort window at runtime, where your users can modify (prior to generating the report) the sort components as to the report's order and component sort type (ascending or descending). Use the update buttons to add the fields that you wish to present to the user here. This sort type is useful when your sort components are fixed, but the user needs to switch the order and sequence occasionally.

Description
Enter a description to use for the field name.

Field
Use the ellipsis to select a field name from the *Select Column* dialog.

Order
Select Ascending or Descending from the drop list, to designate an ascending or descending sort order.

Sorting Window
The components of the sorting window generated at run time can be modified here. In addition to the **Title** text and Status Bar **Message,** buttons can also be customized with a variety of options. Click here for additional help.

DynamicNamed
Valid for Report and Process only. This sort type is similar to DynamicSorting above, but is expanded here to allow groups of sort fields to be referenced by a single name. This is powerful with a generic report designed with no default key. Enter different Names, each one with different sort criteria. A single report can now be used for "on the fly" sorting based on the sort groups that you provide.

Name
Enter a group name to use to identify the sort sequence that you design.

Field
Use the ellipsis to select a field name from the *Select Column* dialog.

Order
Select Ascending or Descending from the drop list, to designate an ascending or descending sort order.

Sorting Window
The components of the sorting window generated at run time can be modified here. In addition to the **Title** text and Status Bar **Message,** buttons can also be customized with a variety of options. Click here for additional help.

BLOBInControl Extension Template

The BLOBInControlExt extension template allows you to display and/or edit a BLOB from a selected table. Normally, BLOB processing involves populating a local or global variable that will load the BLOB contents from a selected table. As the BLOB is updated, the contents of the control is transferred back to the BLOB and stored in the table. This template is designed to handle all of the processing that is necessary to do this.

The use of this template is valid in Browse, Form, or Report procedures to retrieve (read) a BLOB's contents and assign it to a control. In addition, you can also save (write) the BLOB's contents in the Form procedure. A list box is provided to allow you to enter multiple BLOB controls. On the *BLOB Fields Properties* window, the following template prompts are provided:

BLOB Field Select from the drop list the target BLOB field defined in the table that will be displayed and potentially updated (if used on a Form).

Control Select the Field Equate Label, from the drop list, of the control that will be used to display and/or update the BLOB contents. This control must be either a text or image control, depending on whether the type of information contained in the BLOB column is text or Binary (i.e. Image) data. If this is a text control, the use variable for the control must be a string type variable and it must be large enough to hold the largest potential Blob data in the associated Table that you plan to update.

When Image does not exist use the following image – Image File:
If the control is an image control, enter the name of a valid image file (or press the ellipsis button to select one) that will display if an image does not exist in the target BLOB control.

Resize to control size Check this box if you would like the BLOB's contents to resize to fit the image control. If not checked, the default size as stored in the BLOB is used. Using the default could cause the image to display cropped if the image control is too small.

Condition to assign from control to BLOB
Enter an expression in the **Condition** field which will force the BLOB to be updated from the target control *only* when the expression evaluates to TRUE. Press the ellipsis button to select a variable to use in the expression. By default, this setting is TRUE, which means that the BLOB field will *always* be updated when the Form is Accepted, even if the BLOB contents did not change. To improve program performance and network traffic, enter a condition that will cause the BLOB to be updated only when necessary. For example:

```
?Text{PROP:Touched}=True     !was the text field modified?
```
Or
```
CLIP(LOC:ImageFileName)<>'' !was an image loaded into BLOB?
```

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.

The DbAuditing template provides the following prompts:

Log Files

This tab contains a list box that lets you define log files and options for the files used in an application. Add a new log file definition by pressing the **Insert** button. This displays a dialog where you name the log file and define its contents. Use the and to change the order of the log files.

File to Log
Choose a file from the drop list that consists of all files defined in the dictionary. Any updates to this data file are written to its associated log file.

LogFile Name
Specifies the name of the log file to update.

Username Logging

The username logging option is used to track which user is making the update to the specific data file. It does this by adding an extra column to the log file for the user's login name.

Username variable
Specifies the variable which contains the current user. The value of this variable gets written to the log file to indicate the user who made the change to the data file.

Username Header
Specifies the header to be used in the log file for the user column.

Username picture
Specifies the format to be used for the user column in the log file.

File Fields to Log

This option gives the ability to customize a log file for each data file. For example, you may wish to log changes to certain fields and allow changes to others without logging them. Press insert to select a field to write to the log file. Use the and to change the order that the fields are logged to the log file.

Field
Specifies the field name for the data that is written to the log file.

Field Header
Specifies the header to be used in the log file for the data column.

Field Picture
Specifies the format to be used in the log file for the data column.

Miscellaneous Fields to Log

This option gives the ability to customize a log file for miscellaneous data. Press insert to add a field to the log file. Use the  and  to change the order that the miscellaneous fields are logged to the log file.

Field
Specifies the field name for the data that is writen to the log file.

Field Header
Specifies the header to be used in the log file for the data column.

Field Picture
Specifies the format to be used in the log file for the data column.

FileUserTags

This global template extension identifies certain tables and columns that possess a particular **User Options** tag defined in the dictionary. The template then marks or assigns that tag, and allows the programmer to reference the tag *in place of* the physical table or column name. This allows the programmer to write code using the **User Options** tags in the dictionary in place of the actual names used to define them.

The FileUserTags template provides the following prompts:

Interesting File Tags

Specifies defined table tags from the dictionary to use as a reference to the table that contains the tag.

Interesting Field Tags

Specifies defined column tags from the dictionary to use as a reference to the columns that contain the tag.

RunCommandLineProc

The RunCommandLineProc global template extension 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.

The RunCommandLineProc template provides the following prompts:

Log All Errors Silently
Check this box to disable the ErrorManager error windows while running the procedure nominated on the command line. Errors are written to a log (.TXT) file. This avoids the need for any user intervention.

Procedure Name Flag
Specifies a special command line parameter to check for when the program is started. A variable can be specified here if it is preceded by an exclamation point (!).

Call generated procedure
Check this box if the presence of the command line parameter on the command line will cause a procedure generated from the application to be called.

Procedure To Run
Specifies the procedure to be called when the command line parameter is passed to the program on the command line. Choose a procedure from the drop down list if a generated procedure will be called. Enter the name of the procedure in the entry column if it is am external procedure to the application.

Set DLL Image Base Memory Address

This template extension allows you to rebase your application's DLLs. Rebasing is a technique used to modify the base memory address of your DLL.

The base address of a DLL is the preferred location in your application's virtual memory address space where the loader attempts to place the DLL. It is generally specified at link time and used by the linker to write address pointers into the DLL binary . If a DLL cannot load at its preferred base address because the memory region required is already occupied, the loader relocates the DLL elsewhere in virtual memory, then updates all of the address pointers in the DLL to adjust them for the new base address. Making these changes can be time consuming, because the image must be copied to the page file and modified for the new address. This also consumes page file memory until the application is closed and results in the DLL not being shared. A properly based DLL can be demand loaded from disk, so it consumes no page file memory and can be shared. If you have a large application that uses many DLLs, it is important for each DLL to have a different base address to minimize load time and memory use.

It is best to base DLLs from the top of the address range down, instead of from the bottom up. Dynamic memory is allocated from the bottom up and if a DLL tries to load where dynamic memory has been allocated, it will be relocated, just as if a DLL was loaded at that address.

This template is only valid for applications that use DLL as their target output type. EXE and LIB target types are not affected by this template.

Choose from the following prompts:

Select method
Choose *Specify manually*, if you would like to manually assign the DLL's new Image Base Address, or *Choose from list* to provide a valid set of choices to choose from.

Manual Address

This tab contriol allows you to specify a manual image base memory address for your target DLL.

DLL Base Address in hex
Enter an eight digit memory base address in hexadecimal format.

The DLL Base Address must be between 02000000 and 6FFF0000 and end with 0000.

Your EXE and the Clarion libraries use addresses 00400000 to 02000000, so this template will prevent you from using any address below 02000000

Use a Process viewer (like Dependency Walker) to verify that your address space as no conflicts. If you do cause a conflict, the Windows loader will resolve it for you automatically.

IMAGE_BASE ########h will be added to the Export file with the address specified if the target file is a DLL.

Choose Address

This tab control allows you to select a memory address from a list of image base memory addresses to use for your target DLL.

DLL Base Address in hex
Enter an eight-digit memory base address in hexadecimal format.

Select a different base address for each DLL you use in your application.

The listed addresses are on 1 MB boundaries and should allow room for large DLLs. If your DLLs are over 1 MB in size, then skip an address to allow more room.

Use a Process viewer (like Dependency Walker) to verify that your address space as no conflicts. If you do cause a conflict, the Windows loader will resolve it for you automatically.

IMAGE_BASE ########h will be added to the Export file with the address specified, if the target file is a DLL.

Version Resource extension template

The Version Resource extension template gives you the ability to create and implement a Version Information Resource file to use with your application target (executable or DLL).

The information generated by this Version Resource file can be found by right clicking on your executable or DLL and selecting the Version tab.

Version Information is controlled by the Project System. This template automatically generates a version file using the *applicationname.version* format, and includes it in the project. You can learn more about this process by referencing the Version Information Resource Files topic in the *Advanced Topics* PDF.

The following template prompts are provided:

General

Company Name
Enter the name of the company that produced the file (e.g., - *SoftVelocity Corporation*)

Legal Copyright
Optionally enter any copyright notices that apply to the file. Include any pertinent text, symbols, and dates. (e.g., - *Copyright © SoftVelocity 1994-2003*)

Legal Trademarks
Optionally enter any specific trademark or registered trademarks that apply to the file.

Comments
Optionally enter any information that may be used for diagnostic or general information purposes.

Note:

There is a setting in the Version Resource File that is not directly controlled by the templates. This is the "OriginalFilename" VALUE that is set from the value of the Project "Target file".

For example, if your target file is set to "CWHH60.DLL", then the .version file will contain the following line:

```
VALUE "OriginalFilename", "CWHH60.DLL\0"
```

Product Version

Use External Product Information
Check this box if you would like to include the product information from an external source file. Enter a **Source Version File** to include. This file must conform to the proper Version string information standards required by the resource file.

This is useful when compiling multiple applications that are all used with a product theme, like a payroll or accounting package.

Product Name (Required)
Enter the name of the product that is associated with this file. For example, you may distribute a special utility or tool that is associated with an overall accounting software package.

Product Major Version (Required)
Enter an integer that represents the major version of the product that this file is distributed for.

Note:

Product and File Version information use the following format:

major.minor.sub.buildnumber

For example, if major = 6, minor = 0, sub = 1, and build number = 555, the product version results as 6.0.1.555

Product Minor Version
Optionally append a letter or number to the minor version.

Product Sub version
Optionally append an integer to the sub version.

Use Generated Build Number
Check this box to allow the template to generate a build number for you. The template maintains a text file with a .BLD extension that increments the build with each source generation. You can manually edit this file if you wish, or only check this box when you are ready to update versions.

Product Build Number
Enter a valid integer here that represents the build number.

File Version

Internal Name (Required)
Enter the internal name of the file. This could be a module name if the file is a dynamic linked library. In most cases, this name should be the same name as the executable.

File Description (Required)
Enter the name of the file description that the users will see. This name appears directly on the Version tab.

File Major Version (Required)
Enter an integer that represents the major version number of this file. This name appears directly on the Version tab.

Note:

File Version information uses the following format:

major.minor.sub.buildnumber

For example, if major = 6, minor = 0, sub = 1, and build number = 555, the product version results as 6.0.1.555

File Minor Version
Optionally append a letter or number to the minor version.

File Sub version
Optionally append an integer to the sub version.

Use Generated Build Number
Check this box to allow the template to generate a build number for you. The template maintains a text file with a .BLD extension that increments the build with each source generation. You can manually edit this file if you wish, or only check this box when you are ready to update versions.

File Build Number
Enter a valid integer here that represents the build number.

Pre-Release Build
Check this box to mark this file as one that is part of a pre-release build. Although your user will not see anything special generated by this option, it is still useful for developers using the internal Version Info functions during installs and updates. This option primes the internal VS_FF_PRERELEASE flag.

Locale

Language
Select a user's target language in the drop list provided. The language you select will appear in the Version information tab

Code Page
Select a valid Code Page to assign to the Version Information. The default value is *multilingual*.

Clarion

Include Clarion Version Information
Check this box if you would like to include Clarion specific information into the generated Version output. You can override the **Clarion Version, Template family,** and **Template version** information that is extracted by the template.

User

Additional string information that will appear in the Version Info list located on the Version tab can be added here (up to 4 entries). Enter a **label** and corresponding **value** in the template prompts provided.

WindowResize extension template

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.).

The template generates code to reposition the controls, resize the controls, or both, when the end user resizes the window.

Tip

To allow window resizing you must set the WINDOW's frame type to Resizable and you must check the Immediate box to add the IMM attribute to the WINDOW.

Overriding Resize for a specific control

Resize Strategy Specifies the method for resizing and repositioning the controls to fit within the new window size.

Resize

The generated code scales all window coordinates by the same amount, thus preserving the relative sizes and positions of all controls. That is, all controls, including buttons and entry fields get taller and wider as the window gets taller and wider. Window fonts are unchanged.

Spread

The generated code applies the following strategies to the respective control types:

Button
Only the horizontal position (X coordinate) is scaled with the window; width, height, and vertical position are unchanged.

Radio Button
Horizontal and vertical position are scaled with the window, but width and height are unchanged.

Check Box
Horizontal and vertical position are scaled with the window, but width and height are unchanged.

Entry
Width, horizontal and vertical position are scaled with the window, but height is unchanged.

Combo Box
Width, horizontal and vertical position are scaled with the window, but height is unchanged.

Spin Box
Width, horizontal and vertical position are scaled with the window, but height is unchanged.

Drop Combo
Width, horizontal and vertical position are scaled with the window, but height is unchanged.

Other Controls
All coordinates are scaled with the window.

Surface
Makes the most of the available pixels by positioning other controls to maximize the size of LIST, SHEET, PANEL, and IMAGE controls. We recommend this strategy for Wizard generated windows.

Tip

Even though list boxes may be resized, the column widths within the list box are not resized. However, the right-most column does expand or contract depending on the available space.

Don't Alter Controls
Controls are not resized when the window is resized.

Tip

For this strategy, you may add the SCROLL attribute to each control plus the HVSCROLL attribute to the window to provide a 'moving window' over a larger page.

Restrict Minimum Window Size
Check this box to specify a minimum window height and width. This lets you enforce a minimum reasonable size of the window based on the size and number of controls on the window. In other words, you can keep your end user from shrinking the window so much that its controls become invisible or unrecognizable.

Minimum Width
Specify the minimum width of the window in dialog units. Dialog units are based on the window's font and are 1/4 of the average character width.

Zero sets the window minimum to the size at which the window opens (not necessarily the design time size). In other words, it takes into account any .INI setting plus any runtime Property syntax. Thus, we allow the developer to open the window, perform any dynamic control production (including resizing the window) before the minimum restriction takes effect.

Minimum Height
Specify the minimum height of the window in dialog units. Dialog units are based on the window's font and are 1/8 of the character height.

Zero sets the window minimum to the size at which the window opens (not necessarily the design time size). In other words, it takes into account any .INI setting plus any runtime Property syntax. Thus, we allow the developer to open the window, perform any dynamic control production (including resizing the window) before the minimum restriction takes effect.

Restrict Maximum Window Size
Check this box to specify a maximum window height and width. This lets you enforce a maximum reasonable size of the window.

Maximum Width
Specify the maximum width of the window in dialog units. Dialog units are based on the window's font and are 1/4 of the average character width.

Zero sets the window maximum to the size at which the window opens (not necessarily the design time size). In other words, it takes into account any .INI setting plus any runtime Property syntax. Thus, we allow the developer to open the window, perform any dynamic control production (including resizing the window) before the maximum restriction takes effect.

Maximum Height
Specify the maximum height of the window in dialog units. Dialog units are based on the window's font and are 1/8 of the character height.

Zero sets the window maximum to the size at which the window opens (not necessarily the design time size). In other words, it takes into account any .INI setting plus any runtime Property syntax. Thus, we allow the developer to open the window, perform any dynamic control production (including resizing the window) before the maximum restriction takes effect.

Overriding the Resize Strategy for a Specific Control

Window Control
Select a control from the drop-down list.

Disable Resizing for this Control?
Check this box to prevent resizing of the selected control when the user resizes the window. The control will retain its design-time dimensions.

Horizontal Resize Strategy

Specify how the control's width is determined when the end user resizes the window.

Choose from:

Lock Width	The control's design time width does not change.
Constant Right Border	Locks right edge, moves left.

Vertical Resize Strategy

Specify how the control's height is determined when the end user resizes the window.

Choose from:

Lock Height	The control's design time height does not change.
Constant Bottom Border	Locks bottom edge, moves top.

Horizontal Positional Strategy
Specify how the control's horizontal position is determined when the end user resizes the window.

Choose from:

Lock Position	The control's left edge maintains a fixed distance (the design time distance) from parent's left edge.
Fix Right	The control's right edge maintains a proportional distance from parent's right edge.
Fix Left	The control's left edge maintains a proportional distance from parent's left edge.

Fix Center	The control's center maintains a proportional distance from parent's center.
Fix Nearest	Applies Fix Right or Fix Left, whichever is appropriate.

Vertical Positional Strategy
Specify how the control's vertical position is determined when the end user resizes the window.

Choose from:

Lock Position	The control's top edge maintains a fixed distance (the design time distance) from parent's top edge.
Fix Bottom	The control's bottom edge maintains a proportional distance from parent's bottom edge.
Fix Top	The control's top edge maintains a proportional distance from parent's top edge.
Fix Center	The control's center maintains a proportional distance from parent's center.

Fix Nearest
Applies Fix Top or Fix Bottom, whichever is appropriate.

Wise-Generate Wise Installation Script

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

Script Path Specify the path for the Install script. This is specific to this application

Overwrite with default .wse
Check this box to copy the default Wise Script (installed to your \TEMPLATE directory) to your application's local directory (overwriting the local script). Clear the check box if you want to maintain the local script file for this application.

Include External DLLs Check this box if you want the runtime DLLs included in this installation script.

Application Title Specify the name of your application as you want it to appear on the shortcut the installation creates.

Program Group Label Specify the name of the program group you want the installation to create.

Default Folder Specify the folder you want the installation to offer in its "Select Folder" step.

Additional Files to Install
Press this button to add other files (e.g., on-line documentation, data files, etc.). These files are added to the installation script. When adding files, you can also specify whether or not to create a shortcut for each file and the label to display for the item.

Note:

The Wise script this template creates does not list install files in the Wise for Clarion environment.

Progress Window Support (Process/Report)

ExtendProgressWindow

The ExtendProgressWindow template adds functionality to Process and Report procedures. It is designed to do two things:

- Give you precise control over the visual feedback you provide end users for (small) Process and Report procedures.
- Allow Process and Report procedures to operate in two separate modes--all records mode and single record mode (current value range-limit).

You can use the ExtendProgressWindow template to delay or to completely suppress the progress window for a Process or Report procedure, and you can optionally specify a wait cursor. In single record mode, you can suppress the progress window, the print preview, or both.

The ExtendProgressWindow template provides the following options.

Delay Showing Window

Enter the number of seconds to hide the progress window. For example, you may want to hide the progress window for 3 seconds so that processes or reports that finish within 3 seconds limit never show a progress window.

Wait cursor

Check this box to display a wait cursor (hour glass cursor) for the duration of the process or report. For small/short processes and reports, your end users may prefer a simple wait cursor over a progress window. On completion, the procedure restores the cursor to its previous state.

Single Shot

These options are available only for Processes and Reports that specify a key in the **File Schematic** dialog.

Single record Check this box to allow the Report or Process to operate in its normal mode (process all records), or to operate in single record mode (current value range-limit) when GlobalRequest is set to ProcessRecord (see *Procedure Templates--Inter-Procedure Communication* for more information on GlobalRequest).

Tip: If your Report or Process procedure uses a non-unique key, you can process all records with the current key value!

The BrowsePrintButton template primes the range-limit field and calls procedures in this single record mode (see *Control Templates--BrowsePrintButton*).

Use Progress Check this box to display the progress window in single record mode. Clear the box to suppress the progress window in single record mode.

Use Preview Check this box to provide the print preview in single record mode. Clear the box to suppress the print preview in single record mode.

Pause Button Control template

This control template adds a button to allow pausing and restarting of a Process or Report procedure.

Pause Text	The text to display on the button when the process is running. This text alerts the user that the process can be paused by pressing the button. The default is *Pause.*
Restart Text	The text to display on the button when the process is paused and multiple starts are allowed. This text alerts the user that the process can be restarted by pressing the button. The default is *Restart.*
Start Paused	The state of the control when the procedure starts. If check, the process is paused until the user presses the button.
Start Text	The text to display on the button when the procedure opens. This text alerts the user that the process can be started by pressing the button. The default is *Go.*
Allow Multiple Starts	Check this button to allow the process to start more than once.
When Pressed	The standard set of prompts for buttons. Normally, when using a Control template, these prompts are not used.

ProcessReportQBEButton

The ProcessReportQBEButton template places a **Query** button on the progress window for a Process or Report procedure. This button allows the end user to apply a dynamic (run-time) filter to a report. The end user can query the underlying data and print the results of the query in a report. This control template requires a PauseButton control template on the Progress Window to allow the end user the opportunity to press the **Query** button.

The prompts for this control template mirror the template prompts for the BrowseQueryButton control template. See *Read-Write Browse Templates—BrowseQueryButton* for the deatiled information on these prompts.

The end user may provide filter criteria for zero or more columns. Additional filter criteria result in a more refined search and a smaller result set (the filter conditions are conjunctive—ANDed together).

Runtime Options

The default comparison operator is (=), which searches for an exact match between the report target field and the corresponding Query input field. By default all matches are case sensitive. Pressing the comparison operator button cycles through all the available operators:

Operator		**Filter Effect**	
=	*browsefield*	equal	*queryvalue*
>=	*browsefield*	greater than or equal	*queryvalue*
<=	*browsefield*	less than or equal	*queryvalue*
<>	*browsefield*	not equal	*queryvalue*
	no filter		

For string fields, you may use the following special characters in the Query input field to refine your search:

Symbol	**Position**	**Filter Effect**		
^	prefix	caseless (case insensitive) search		
*	prefix	*browsefield*	contains	*queryvalue*
*	suffix	*browsefield*	begins with	*queryvalue*

For example:

d - matches 'd' only
d* - matches 'dog', 'david'
*d - matches 'dog', 'cod'
^*d - matches 'dog', 'cod', 'coD'

Upon completion of the Query dialog, the current sort order of the report is filtered to match the query. If Query is selected again, the previous query is available by default. This allows sharing of filters between sort orders, as well as successive filter refinements.

The standard Query behavior is defined by the ABC Library's QueryClass. See *QueryClass, QueryFormClass,* and *QueryFormVisual* for more information.

The ProcessReportQBEButton template provides the following prompts:

General

QBE Family

Specify the query family for the QBE interface. Normally this is the name of the procedure.

QBE Interface
Select the query interface from the drop-down list. Choose from

Form One input field and button per Query field

List One listbox row per Query field

Auto Populate
Check this box to provide a query dialog with filter criteria for each field in the report. The input fields have the same picture token and prompt as the corresponding report field.

Clear this box to enable the **Fields** button and specify custom query input fields.

Caseless Auto Populate
Check this box to do case insensitive searches for each field in the report target. Clear the box to do case sensitive searches.

Fields
Press this button to populate specific query input fields. You can use this option to restrict the query to some subset of report fields, or to expand the query to include fields not in the report. You can also implement caseless searches by default.

Field

Type the field name to include in the Query dialog, or press the ellipsis button to select the field from the **Select Field** dialog.

Title

Type the prompt or label associated with the Query field.

Picture

Type a picture token for the Query field, or press the ellipsis button to select a token with the **Edit Picture** dialog.

Caseless

Check this box to do case insensitive searches on the Query field. Clear the box to do case sensitive searches.

Disable Begins\Contains?

Check this box to disable the Begins\Contains queries. This is available for the QBE form interface for any non-string field.

Retain Query

This option is checked by default, and indicates that the end-user's query will remain in the Query dialog on the subsequent press of the Query button. Clear the check box to reset the Query dialog on each press of the Query button.

Use on startup

Check this box to open the Query dialog before the Report procedure opens.

Result Control

Optionally select a STRING Control from the Droplist to display the filter statement created by the QBE object. A property assignment is made to the selected control (using PROP:Text), therefore it is not necessary to associate a variable with the STRING.

Auto-share between tabs
Not available and always disabled in this template.

Quick QBE Support?
Not available and always disabled in this template.

Submenu Icon
Not available and always disabled in this template.

Query Icon
Not available and always disabled in this template.

QBE Class

Select this tab to override the global Query Manager setting. See *Template Overview—Classes Tab Options—Global* and *Local*.

QBE Visual Class

Select this tab to override the global Query Manager setting. See *Template Overview—Classes Tab Options—Global* and *Local*.

Query Center and QBE

BrowseQueryButton

The BrowseQueryButton template provides a **Query** button to let the end user apply a dynamic (run-time) filter to the BrowseBox result set. In other words, the end user can query the underlying dataset and display the results of the query in the BrowseBox list.

The default query interface is a dialog with an input field and a comparison operator button for each list box column.

The end user may provide filter criteria for zero or more fields. Additional filter criteria result in a more refined search and a smaller result set (the filter conditions are conjunctive--ANDed together).

Runtime Options

The default comparison operator is (=), which searches for an exact match between the BrowseBox field and the corresponding Query input field. By default all matches are case sensitive. Pressing the comparison operator button cycles through all the available operators:

Operator	Filter Effect		
=	*browsefield*	equal	*queryvalue*
>=	*browsefield*	greater than or equal	*queryvalue*
<=	*browsefield*	less than or equal	*queryvalue*
<>	*browsefield*	not equal	*queryvalue*
	no filter		

For string fields, you may use the following special characters in the Query input field to refine your search:

Symbol	Position	Filter Effect		
^	prefix	caseless (case insensitive) search		
*	prefix	*browsefield*	contains	*queryvalue*
*	suffix	*browsefield*	begins with	*queryvalue*

For example:

d - matches 'd' only
d* - matches 'dog', 'david'
*d - matches 'dog', 'cod'
^*d - matches 'dog', 'cod', 'coD'

Upon completion of the Query dialog, the current sort order of the BrowseBox is filtered to match the query. If Query is selected again, the previous query is available by default. This allows sharing of filters between sort orders, as well as successive filter refinements.

The standard Query behavior is defined by the ABC Library's QueryClass. See *QueryClass, QueryFormClass, QueryFormVisual, QueryListClass, and QueryListVisual* for more information.

The BrowseQueryButton template provides the following prompts:

General

QBE Family

Enter the family name that this QBE object will use. Normally, the QBE family will default to the procedure name. However, there may be situations where you would like saved queries to be shared between one or more procedures. In this case, use a common family name between these shared procedures.

QBE Interface

Select the query interface from the drop-down list. Choose from

Form	One input field and button per Query field
List	One list box row per Query field

Auto Populate

Check this box to provide a query dialog with filter criteria for each field in the BrowseBox. The input fields have the same picture token and prompt as the corresponding BrowseBox field.

Clear this box to enable the **Fields** button and specify custom query input fields.

Caseless Auto Populate

Check this box to provide a query dialog where the filter criteria will not be case sensitive for each field in the BrowseBox.

Fields

Press this button to populate specific query input fields. You can use this option to restrict the query to some subset of BrowseBox fields, or to expand the query to include fields not in the BrowseBox. You can also implement caseless searches by default.

Field Type the field name to include in the Query dialog, or press the ellipsis button to select the field from the **Select Field** dialog.

Title Type the prompt or label associated with the Query field.

Picture Type a picture token for the Query field, or press the ellipsis button to select a token with the **Edit Picture** dialog.

Caseless Check this box to do case insensitive searches on the Query field. Clear the box to do case sensitive searches.

Disable Begins/Contains?
Check this box to disable Begins/Contains queries. This is available for the QBE form interface for any non-string field.

Retain Query

This option is checked by default, and indicates that the end-user's query will remain in the Query dialog on the subsequent press of the Query button. Clear the check box to reset the Query dialog on each press of the Query button.

Use on startup

Check this box to open the Query dialog before the Browse procedure opens.

Result Control

Optionally select a STRING Control from the Droplist to display the filter statement created by the QBE object. A property assignment is made to the selected control (using PROP:Text), therefore it is not necessary to associate a variable with the STRING.

Auto-share between tabs

Check this box to make the query active to all tabs associated with the browse.

Quick QBE Settings

Check the **Quick QBE Support** to enable special popup menu support for saved queries. A *View* menu item will appear in the Browse box popup menu, with a sub-menu of saved queries. You can customize the *View* menu popup icon by modifying the **Submenu Icon** prompt. Customize the saved query items by modifying the **Query Icon** prompt.

QBE Class

Select this tab to override the global Query Manager setting. See *Template Overview--Classes Tab Options--Global* and *Local*.

QBE Visual Class

Select this tab to override the global Query Manager setting. See *Template Overview--Classes Tab Options--Global* and *Local*.

BrowseQBEList Control Template

The Browse QBEList Control Template (formerly Soft Velocity Query Center) is a special control template that allows you to test table queries easily and view the results. You can drag columns from a browse box into the query area, enter values, and quickly test your data set.

This template requires that a Browse Box be populated on the window. Although there are no special prompts associated with this template, there are multiple controls that are populated together and are described below:

Filtering Center List Box

This is an EIP (edit-in-place) list box that allows you to enter a column name, operator, Value or Expression, and an optional connecting operator to the next query.

You can also drag a column name from the browse box to this list box.

You can also drag specific value from a selected row of the browse. For example, if you select the Column *Name* and drag from a row where the value of that column is *"Bob"*, the QBE List will use that column/value to create *Name = 'Bob'*.

Note:

In the Value or Expression line, do not enter a single quote in your search string. Instead, if you wish to search for 'FL' as your search string, enter ''FL''.

Updates to the browse box forces an automatic refresh of the active query.

Case sensitive search for string

Check this box to force case sensitive string searches for the current active query.

Reset Button

Clears the list box for a new query.

Save Query Button

Saves the query (to the non-volatile storage source specified in the application's global settings).

Note:

Storage is automatic with ABC templates, but you must specify a storage source in the Global Properties of your application when using the Clarion template chain.

Load Query Button

Loads a previous query from the non-volatile storage source.

Apply Button

Executes the current active query.

Browse QBE List Template Prompts

The following template prompts are provided:

Filter class

The base class used for the QBE List. The default is **cFilterList**.

Filter object name:

The object name used for this procedure. The default name is **FilterObj**.

Copy generated filter string to clipboard

Check this box to enable the filter string generated by the Filter Class to be copied to the Windows Clipboard. Pressing the Apply button initiates this feature.

Search on string is case sensitive

Check this box to allow the QBE List to default to case sensitive searches.

Case sensitivity search can be set at runtime

Check this box to allow the **Case sensitive search for string** check box to be populated at runtime that allows a user to enable or disable case sensitive string searches.

Query Center Translation File (cFilterList.TRN)

The Query Center Translation File, *cFilterList.TRN*, is the backbone of the BrowseQBEList control template for standard and ADO based browse procedures. It defines all regular and exception group information for both standard and ADO types of applications.

The format of the expression to be used with the associated operator defines “keyword”. Those keywords are enclosed in square bracket ([]) and will be replaced at runtime by the appropriate value.

Those Keywords are:

```
[COL]
[VAL]
[V1]
[V2]
[MV]
[MV, Picture]
[APP_PIC]
[PIC, Picture]
```

Some of these keywords have a direct relation with the BrowseQBE Advanced interface. For example, in the query interface:

Data Filter Center

Column	Operator	Value/Expression	Connecting Operator
Customer ID	Equal	AROUT	DONE

The keyword **[COL]** will refer to the list column called “Column”. The **[VAL]** keyword will refer to the list column “Value/Expression”.

When the “Apply” button is pressed, the code executes the following steps.

1. It will loop through all the lines in the Filter list.
2. For each line, the code is doing a lookup in an Exception queue to see of there is an exception handler for the operator and the data type involved. In this case, there is no special handling for that data type so the code will use the entry in the standard group for “Equal”

```
pstring('Equal')
pstring('[COL] = [VAL]')
```

3. The internal queue has the information that "Customer ID" corresponds to a column SQL name of Customer.CustomerID. So the code replaces the keyword **[COL]** by "Customer.CustomerID". At this point, the filter string will look as:

```
Customer.CustomerID = [VAL]
```

4. After that, **[VAL]** is replaced by the content of "Value/Expression" column of the current filter line that is under process. If there is the need for it, quotations around the value will be added. So the final filter string generated by the code will be:

```
Customer.CustomerID = 'AROUT'
```

[V1] and **[V2]**

These keywords tell the code that content of "Value/Expression" column contains 2 values separated by a comma. Example of operators that involve this format is BETWEEN, NOT BETWEEN as defined in the group:

```
pstring('Between')
pstring('INRANGE([COL], [V1], [V2])')
```

[MV]

This keyword means that the content of "Value/Expression" is representing 2 values or more and the code will generate a set. For example, given the following group entry for SQL:

```
pstring('IN')
pstring('[Col] IN ([MV])')
```

and using the following values from the application:

Data Filter Center

Column	Operator	Value/Expression	Connecting Operator
Customer ID	IN	ALFKI, ANATR, ANTON	DONE

The code will generate the following filter string:

```
Customers.CustomerID IN ('ALFKI','ANATR','ANTON')
```

[MV, Picture]

This variant of the [MV] keyword will indicate to the code that the values returned in the set should be formatted using the picture specified. If necessary, the code will deformat the value before applying the needed picture.

For example, the cFilterList.TRN has the following entry in the MSSQLDateTime section:

```
ulong(datatype:date)
pstring('IN')
pstring('CONVERT(char(15), [COL], 111) IN ([MV,@d10])')
```

If the Value/Expression column contains ‘21/09.2003, 22/09/2003’, the result filter string will be:

```
CONVERT(char(15), ColDate, 111) in (‘2003/09/21’, ‘2003/09/22’)
```

Specialized keywords

[APP_PIC], **[PIC, Picture]**

On many occasions, generic processing is not enough and a more specialized processing is needed. A good example for this is date processing. The expression the code will use to build the filter string is using DEFORMAT for date processing and that statement involves a picture (ex: @d10). This is where the **[APP_PIC]** keyword finds its duty; it supplies to the code a placeholder where that data picture used in the app will be inserted in the expression.

The keyword **[PIC, Picture]** is always used in pair in the expression with **[VAL]**, **[V1]** or **[V2]**. It simply forces the code to format the value using the picture supplied by the second argument of **[PIC, Picture]**. For example, the following group entries from the MSSQL exception group use that keyword.

```
ulong(datatype:date)
      pstring('Equal')
      pstring('CONVERT(char(15), [COL], 111) = ''[PIC, @d10][VAL]''')
```

This means that if the application is using ADO and the backend is MSSQL, an expression in the Filter list that uses a date and the equal operator will see the generated string as follows:

```
CONVERT(char(15), Employee.DateHired, 111) = ‘1994/10/22’
```

Relation Tree Template Support

RelationTree control template

Click on a TAB to see its help

The tree control is a list box formatted to display as a collapsible hierarchical list. This Control template provides an alternative for the Browse-Form paradigm. A single RelationTree control can replace several Browse-Form pairs.

Using the RelationTree Control template, you can specify multiple related files to display on multiple levels (up to 29) of a hierarchical list--*with an associated update procedure for each level*. The related files are declared in the File Schematic--the Primary (Parent) file and a single chain of related secondary Child files (Parent-Child-GrandChild).

The RelationTree template employs a fully-loaded QUEUE for the root level. The child levels are demand-loaded when a branch is expanded.

Tip

This template is not appropriate for databases with a very large primary file. For large files you should use the BrowseBox Control template.

The plus (+) sign indicates a collapsed level that expands when the user CLICKS on the plus (+) sign. Conversely, the minus (-) sign indicates an expanded level that collapses when the user CLICKS on the minus (-) sign.

To create a tree using the RelationTree Control template:

1. Place a RelationTree Control template on a window.

   This opens the **List Box Formatter**. Use the **List Box Formatter** to enable colorization, icon display, or horizontal scrolling in your tree control (see *The List Box Formatter*). Do *not* use the **List Box Formatter** to populate fields in the tree control.

Tip

The tree control is a single column list, therefore you must specify a column scroll bar rather than a list scroll bar to accomplish horizontal scrolling.

2. Press the **OK** button on the **List Box Formatter**.
3. RIGHT-CLICK on the RelationTree Control template and choose **Actions** from the popup menu.

4. Press the **Files** button to specify the file schematic for the control.

   Specify the Primary (Parent) file and a single chain of related Secondary Child files (Parent-Child-GrandChild).

5. Complete the RelationTree template prompts.

The RelationTree template provides the following prompts:

File Details

Tree Heading Text

An optional text heading at the top of the tree. Tree Heading Text is required to let the user add a record at the root level.

Tree heading Icon

An optional icon at the top of the tree. Icons must be enabled in the List Box Formatter for this prompt to be enabled.

Expand Branch

Specify a keystroke to expand the selected list item--display its children. Press the ellipsis button (...) to select special keys such as ESC, TAB or ENTER. See *Controls and Their Properties--Common Control Attributes--Setting the KEY Attribute* for more information on this dialog.

Contract Branch

Specify a keystroke to contract the selected list item--hide its children. Press the ellipsis button (...) to select special keys such as ESC, TAB or ENTER. See *Controls and Their Properties--Common Control Attributes--Setting the KEY Attribute* for more information on this dialog.

Accept control from Toolbar

Check this box to accept navigation events and other relation tree control events generated by the *FrameBrowseControl* control template on the APPLICATION's toolbar. See *FrameBrowseControl* for more information on these toolbar buttons and their operation. Clear this box to disable the *FrameBrowseControl* toolbar buttons for this procedure.

Give option to expand and contract all levels

Specify the RIGHT-CLICK popup menu for the RelationTree includes "Expand All" and "Contract All" commands.

Primary File Settings

Display String

The field name or text to display for the primary file level. This may be any valid Clarion expression, for example:

```
CLIP(CUST:LastName)&' '&CUST:FirstName
```

Update Procedure

The update procedure to call for the primary file. The procedure may be accessed with the RIGHT-CLICK popup menu automatically provided when you specify an update procedure. The default popup menu text is "Insert," "Change," and "Delete."

The procedure may also be accessed with the RelationTreeUpdateButtons--see below. If you use the RelationTreeUpdateButtons control template, the popup menu inherits the text from the buttons.

Record Filter

Type a valid Clarion expression to limit the contents of the list to only those records causing the expression to evaluate to true (nonzero or non-blank). The procedure loops through all displayable records to select only those that meet the filter.

You must BIND any file field, variable, or EQUATE that is used in a filter expression. The **Hot Fields** tab lets you BIND fields.

Colors

This tab is only available if you check the **Color Cells** box in the **List Field Properties** in the List Box Formatter.

Default Colors

To specify the default colors for the primary file display string, type color EQUATEs (from \LIBSRC\EQUATES.CLW) in the entry fields or press the ellipsis (...) buttons to select colors from the **Select Color** dialog.

Conditional Color Assignments

To specify conditional colors for the primary file display string, press the **Insert** button. This opens the **Conditional Color Assignments** dialog.

Conditional Color Assignments

This dialog lets you specify the conditional colors for the primary file display string.

Condition

Type a valid Clarion expression to evaluate at runtime, then type color EQUATEs (from \LIBSRC\EQUATES.CLW) in the entry fields or press the ellipsis (...) buttons to select colors from the **Select Color** dialog.

At run-time these conditions are evaluated, and the colors for the first true condition in the list are used.

Icons

This tab is only available if you check the **Icons** box in the **List Field Properties** in the List Box Formatter.

Default Icon

To specify the default icon for the primary file display string, type the icon filename in the entry field.

Conditional Icon Usage

To specify conditional icons for the primary file display string, press the **Insert** button. This opens the **Conditional Icon Usage** dialog.

Conditional Icon Usage

This dialog lets you specify conditional icons for the primary file display string.

Condition

Type a valid Clarion expression to evaluate at runtime.

Icon

Type the icon filename in the entry field.

At run-time these conditions are evaluated, and the icon for the first true condition in the list is used.

Secondary File Settings

The secondary file settings are identical to the primary file settings. Highlight the secondary file, then press the **Properties** button below the **Secondary Files** list box. See *RelationTree Overview* for information on how to specify the secondary files with the **Select File** dialog.

RelationTree Embed Points

The RelationTree Control template provides a comprehensive set of embed points to allow full customization of the control's behavior.

Relation Tree Expand/Contract Buttons control template

This Control template adds two buttons (**Expand** and **Contract**) which let the user expand or contract all the items in the relation tree. A Relation Tree control template. must be present on the window. There are no prompts for this template.

Relation Tree Update Buttons control template

This Control template adds three buttons (**Insert**, **Change**, and **Delete**) which allow the user to call the associated update procedure for the selected level of a Relation Tree (if an update procedure has been specified) . There are no prompts for this control. The Update Procedure is specified for each level of the Relation Tree control template. .

The Change and Delete buttons correspond to the currently highlighted record. The Insert button adds a child record (the next level down the tree structure).

Report Based Templates

ReportChildFiles

The ReportChildFiles template adds functionality to Process and Report procedures. This extension template provides a simpler, more efficient, more controllable alternative to setting a chain of related files in the File Schematic and having the Report or Process template produce a single multi-tiered VIEW.

The ReportChildFiles template lets you name only the primary file and any lookup files in your procedure's File Schematic. The template generates code to read (and optionally print a separate DETAIL for) the related child-file records for each primary file record. We recommend the ReportChildFiles template for the typical invoice headers/invoice lines scenario.

Multi-tiered View

Suppose you have an invoice header file and an invoice detail file. You want to print out a header and then a line for each detail. This is somewhat tricky to do with a single view and there are some limitations and inefficiencies with this approach. You must populate each header (parent) file field into a group HEADER and each detail (child) field into a DETAIL. The limitation is there are no events and no embed points to use when the parent record prints (because it is simply a group break). The inefficiency is that additional GETs are done on parent file lookups for every child record even though the parent record is unchanged. Plus, for SQL you must use a left outer join (inefficient) to force parent headers to print when there are no associated detail lines.

ReportChildFiles

With the ReportChildFiles template you can simply populate the header (parent) as the primary file with its own DETAIL, then populate a second DETAIL for the detail (child) file. The primary view is then read record-by-record (lookups done only once for each parent record) and the child view is range-limited on the parent file linking fields. The Process Manager Method TakeRecord embed point provides an access point for *both* parent and child records. ProcessClass.TakeRecord is called for each record (parent or child), and ProcessClass.ChildLevel indicates which file/record is active. See *ProcessClass* for more information.

Using the ReportChildFiles Template

The ReportChildFiles template provides the following options.

Parent File

Type the label of the parent file, or press the ellipsis button (...) to select the parent file from the **File Schematic** dialog.

Detail

For Report procedures, select the USE attribute (field equate label) of the REPORT DETAIL structure to print for each child record.

Tip: The Detail drop-down list shows DETAIL structures with USE attributes, so populate the DETAIL first, and add a USE attribute.

File Schematic <To Do>

Insert the *child* file to process for each parent file record.

Classes Tab

Use the Classes tab to override the global ViewManager setting. See *Template Overview--Classes Tab Options--Global* and *Local*.

ReportDateStamp control template

The ReportDateStamp template adds two STRING controls to a REPORT: a "Report Date:" text STRING, and a formatted variable STRING to display the date. By default, the ReportDateStamp template displays the system date using the Windows standard long date format (D18). For example, November 18, 1997. However, you may select an alternative format and an alternative date value to display.

The ReportDateStamp template provides the following prompts:

Format Picture

Press the ellipsis button to select a date format. See *Picture Tokens* in the *Language Reference*.

Use System Clock?

Check this box to display the system date (see *TODAY* in the *Language Reference*). Clear the box to display a variable containing the date value to display.

Date Variable

Type the variable name or press the ellipsis button to select the variable from the **Select Fields** dialog.

ReportPageNumber control template

The ReportPageNumber template adds a variable STRING to display the page number.

The ReportPageNumber template provides no configuration prompts.

ReportTimeStamp control template

The ReportTimeStamp template adds two STRING controls to a REPORT: a "Report Time:" text STRING, and a formatted variable STRING to display the time. By default, the ReportTimeStamp template displays the system time using the Windows standard long time format (T8). For example, 12:90:22 PM. However, you may select an alternative format and an alternative time value to display.

The ReportTimeStamp template provides the following prompts:

Format Picture

Press the ellipsis button to select a time format. See *Picture Tokens* in the *Language Reference*.

Use System Clock?

Check this box to display the system date (see *CLOCK* in the *Language Reference*). Clear the box to display a variable containing the time value to display.

Time Variable

Type the variable name or press the ellipsis button to select the variable from the **Select Fields** dialog.

Report Wizard utility template

This wizard creates a Report Procedure from an existing dictionary file definition.

After the introduction screen, you are presented with the following options:

Theme Selection

Theme	Select from the drop list of themes. Themes are groups of settings that control colors, fonts, icons, backgrounds, positions and much more - for Frame, Browse, Form and Report procedures. You will have the opportunity to create a new theme as you progress through the wizard. Select a starting or default theme here.
Save Settings	After you have selected a theme, you have the option to save these settings for any future applications that you create.

What name should be used as the label of the report procedure?
Type the procedure name.

Which file do you want to report?
Press the ellipsis (...) button to select a file from the dictionary.

Key Sequence - Select Sort Order
Select from the drop list the sort and report generation method from the following choices:

Single Key
Select this option to force the wizard to generate a seperate report for the key that you select in the **Enter a key** prompt that follows.

Runtime Key Selection
Select this option to force the wizard to generate a single report that pops up a sort order dialog prior to printing at runtime.

Record Order
Select this option to force the wizard to generate a single report sorted by record order for your selected file (or files).

How many columns do you want the report to use?
Type the number of columns for your report. The Report Wizard distributes the report columns evenly across the columns.

Select the fields that you want to use

Build your report in this list box by adding and deleting fields from the selected file. You can also modify the properties of the fields regarding column labels, picture tokens, and justification. Use the arrow buttons to specify the order that each field will appear on the report.

Customization

Wizards have different "look and feel" settings and actions called *themes*, which can be modified and saved for use in other applications. Themes are set and controlled by a variety of customization options.

Press the **Next** button to accept the selected theme's settings, or press the Report Customization button to modify it at this time.

Overwrite existing procedures

Check this box to overwrite existing procedures with the same names. Clear the box to preserve existing procedures.

On the last dialog, the **Finish** button is enabled. If you are satisfied with your answers, press the **Finish** button. You also have the option here to **Save Changes**, where any changes to customization options are saved to the theme that you selected at the start of the wizard. If you wish, you can opt to **Save to a new theme.**

The Report Procedure Wizard creates the procedure based on the dictionary table and the answers you provided, and then displays the **Procedure Properties** dialog for your new procedure.

RTF Text Control Support

RTFTextControl Control Template

The RTFTextControl control template provides built-in RichEdit support for native TEXT controls. In previous releases, support for Rich Text Format was provided in a special and detailed set of special support templates. This new RTFTextControl allows for easy integration of RTF support into all of your applications, with additional code and control template support through the RTFToolbar, RTFStatusBar, and the RTFAction Code Template.

In essence, this template simply adds a TEXT control to your window, with the RTF attribute applied. It also uses a *Field* as the RTF **Value Mode** parameter.

Tip

The normal design flow of the new RTF text control support is as follows:

1. Populate the RTFTextControl
2. Populate the RTFToolbar
3. Populate the StatusBar
4. Add the RTFAction Code template to additional control and event embed points if needed.

Classes

The Classes tab lets you control the classes (and objects) your procedure uses to accomplish the template's task—that is, they override the global class settings specified in the **Global Properties** dialog. You may accept the default Application Builder Class specified in the **Global Properties** dialog (recommended), or you may specify your own or a third party class to override the default setting. 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 specified in the **Global Properties** dialog. 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*).

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.

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.

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.

RTFToolBar Control Template

This control template requires the population of the RTFTextControl Control Template.

The **RTFToolBar** Control Template populates a group of support controls to use with the RTFTextControl control template.

There are no template prompts for this control. You can disable or hide controls that you do not want to provide to the user. You can also modify the icons and text used on each support control.

Each control is described as follows:

 **New** Clears the contents of the RTF Control.

 **Open** Prompts you to load an external RTF file's contents into the RTF control

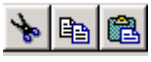
 **Save** Allows you to save the existing RTF control's contents to an external file.

 **Print** Prints the RTF control contents.

 **Find** Opens the *Find* dialog window, allowing you to search for text in the RTF control.

 **Find and Replace**

Opens the *Find and Replace* dialog window, allowing you to search and replace target text in the RTF control.

 **Cut, Copy and Paste**

Performs the standard Windows cut, copy and paste actions for selected text in the RTF control.

 **Undo and Redo**

Allows you to *undo* the sequence of actions performed on an RTF control, and to *redo* those actions if needed.

 **Tab Stops**

Press this button to access the *Tabs* dialog. Set your tab stops in millimeters, inches, or twips. The hotkey to move to the next tab stop is **CTRL+TAB**

Paragraph Indents

Press this button to access the *Paragraph Indents* dialog. Set the **First** line, **Left** side and **Right** side of the paragraph in millimeters, inches, or twips.

Alignment Controls

These buttons provide the standard alignment features (left, right, center, and full justification) for the selected text.

Note:

Full justification is currently only supported in Rich Edit versions greater than 3.0. The presence of RichEdxx.DLL is required in the Windows *System* directory.

Bullets and Numbering/Bullet Styles

Add bullets or numbers and set the style with these buttons. Bullets and numbering can only be applied at the *start* of a paragraph.

Font and Colors

These controls set all aspects of the active font and color to use in the RTF control.

Hiding or Removing Controls

If there are controls that you do not need in your application (Example: *Save* or *Print*), you can simply delete them. When you are prompted with “Do you want to delete this control template?” Press **No**, and only the individual control will be removed.

To conditionally hide controls at runtime, you can use the following embedded source in the Window Manager’s Init Method (just *after* the first RTFControl Initialization embed point):

```
IF NeedToHideSave          !locally defined flag
    HIDE(?RTFToolSave)     !hide Save button
    RTFControl19.AddItem(RTFToolbar:CtlButtonSave,0)
    !the AddItem method disables Save button logic
END
```

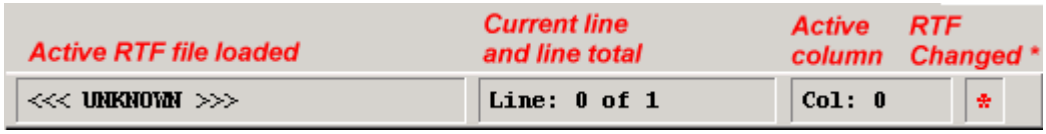
Use this technique for any button that you need to conditionally display.

RTFStatusBar Control Template

This code template requires the population of the RTFTextControl Control Template.

The **RTFStatusBar** Control Template populates a simulated status bar made up of panel and string controls that can be applied to any window.

There are no template prompts for this control. The information contained in this template is shown below:



RTFAction Code Template

This code template requires the population of the RTFTextControl Control Template.

The **RTFAction** code template incorporates special support features for a target TEXT control that is using the RTF attribute. Use this code template with visual controls that identify the action that will be performed (e.g., a Save button to perform a *Save* action).The following Actions, and supported prompts, are documented as follows:

Action

The following actions are supported through the RTFAction code template. Each action provides additional prompts that populate the appropriate methods generated by the underlying class.

Save

This action writes the contents of the RTF control to a designated target. The drop list **Mode** specifies whether to write the contents in *RichText* or *PlainText* format. Your default can be either one of these formats, but if you haven't set the default mode in hand code, the default mode specified by the templates is *RichText* format.

The **File Name** sets the Save destination. This must be a CSTRING variable that you have defined.

Load

The Load action is used to load the contents of an external RTF filename to the target RTF control. The **File Name** sets the target to load. This must be a CSTRING variable that you have defined.

FindandReplace

This action is used to launch a search and replace into a designated RTF control. The **Find** value is searched, and the value in the **Replace** by prompt replaces the **Find value** text**.**

SelectedText

The **SelectedText** action is used to move the current selected text to an external file name designated in the **File Name** prompt.

SetFocus

The **SetFocus** action is used to simply select an RTF control on a window. This could be useful in windows with multiple tab controls. A button could be used to "hot select" the RTF control.

Print

This action is used to print the current contents of an RTF control. Enter a string or variable name to send the Windows **Job Name** to the Print Engine. Make sure to prepend an exclamation point to the variable name (Example - *!MyPrintJobName*)

Touch
The **Touch** action is used like Prop:Touched to generate an Accepted event on the RTF control. You can make the action variable by entering a valid expression in the **Expression** prompt. The default value is TRUE.

Changed
The **Changed** action is used to trigger the other supporting controls (like the status bar and toolbar controls) that a change has occurred, and will prompt the user to save any changes if they have not saved it yet. A value of TRUE (1) will be sent to the variable designated in the **Variable prompt.**

Because rich text can include hidden information, the **Changed** Action can return TRUE even if there are no visible changes, and no calls to Touch(TRUE) have been executed.

Special Purpose Templates

cwRTFGlobal extension template

Note:

This template set is now deprecated in Clarion 6, and is replaced by the more powerful RTFTextControl templates. See the *Templates by Topic* PDF for more information.

The cwRTFGlobal extension template is a requirement for any application that will utilize Clarion's Rich Text support. This extension simply includes the necessary equates and Classes needed to compile the application.

Requirements

This global extension template has no template requirements.

Populating the Template

1. **Press** the Global button from the IDE.
2. **Press** the Extensions button.
3. **Press** the INSERT button and **select** the cwRTFGlobal extension.

Template Prompts

This template has no prompts.

RTFControl control template

Note:

This template set is now deprecated in Clarion 6, and is replaced by the more powerful RTFTextControl templates. See the *Templates by Topic* PDF for more information.

The RTFControl control template adds a TEXT control to a window. The necessary code is generated and used to manipulate the text within the control as Rich Text.

Clarion's implementation of Rich Text allows the rich text field to be saved to a STRING, MEMO, or a separate text file. The minimum size of a STRING or MEMO field is 256 characters. This is needed in order to maintain the rich text header information. If the field contains a lot of formatting a larger field size will be needed.

The components of a rich text control include a Toolbar, Format Bar, and Ruler. These components of the rich text control are optional. They may be shown or hidden at runtime. They default to showing at runtime. The bars may not be modified at design time in the window formatter. When the control is placed on the window, it is a basic TEXT control.

Toolbar
The Toolbar provides basic file access, print, search, clipboard operations and save and redo functionality. Any of these operations may be duplicated with the provided code templates.

Format Bar
The Format Bar provides basic text formatting actions. This includes Font, Point Size, Bold, Italics, Underlining, Color, Paragraph justification and bulleting. Any of these operations may be duplicated with the provided code templates.

Ruler
The Ruler provides a way to measure the width of the text. Tab stops may be set on the ruler by clicking on the ruler. To remove a tabstop, click on an existing one.

Requirements

This control template requires the cwRTFGlobal global extension.

Populating the Template

Open the Window Formatter for the procedure that will contain the RTF control

1. **Select** Populate ▸ Control Template.

   The Select Control Template dialog displays.

2. **Highlight** *RTFControl* and **press** the Select button.
3. **Click** on the window at the position where you want the RTF text control to be placed.
4. **Modify** the size of the control as desired.
5. **RIGHT-CLICK** on the TEXT box, and **select** Actions to set the RTFControl options.

Template Prompts

The RTFControl template provides the following prompts:

Load From File
Specify the .RTF filename to use, or a variable that holds the filename. The .RTF file is used to store the rich text. The variable must be preceeded by an exclamation point (!).

Specify a numeric value to use to limit the amount of text that can be placed into the .RTF file.

Load From Field
Specifies a column in a table that is used to store the rich text data. Use the ellipsis (…) to select the column from the table or type in a column name that already exists in the table. If this option is used, RTF text can only be stored using a STRING or MEMO data type.

Limit RTF Size to match field
Check this box to limit the amount of text that can be placed into the rich text control. The limit of the text will be equal to the size of the field. Clear this box in order to not validate the size of the field.

Check status on window close
Check this box to receive a warning if the text in the rich text control has been modified but not saved. Clear the box for no warning to appear.

Display Ruler Bar
Check this box to default the ruler bar on when the window opens. Clear the box to default it off when the window opens.

Display Tool Bar
Check this box to default the tool bar on when the window opens. Clear the box to default it off when the window opens.

Display Format Bar
Check this box to default the format bar on when the window opens. Clear the box to default it off when the window opens.

HTMLHelp Templates

cwHHGlobal extension template

The cwHHGlobal extension template is a requirement for any application that will utilize Clarion's HTML Help support. This extension simply includes the necessary equates and Classes needed to compile the application and defines the applications default help file.

Requirements

This global extension template has no template requirements.

Populating the Template

1. **Press** the Global button from the IDE.
2. **Press** the Extensions button.
3. **Press** the INSERT button and select the cwHHGlobal extension.

Note: Depending on the Application Template your application uses (Clarion or ABC) the cwHHGlobal extension will be available beneath the Class cwHH – HTML Help for Clarion or Class cwHHABC – HTML Help for Clarion.

Template Prompts

The cwHHGlobal template extension provides the following prompts:

Default Help File Name
Specify the name of the application's compiled help (.chm) file. The filename should not be enclosed in quotes. A variable name can be used, but you must prepend an exclamation point to the variable name (i.e., *!HelpFileName*).

Append .HTM to Help Ids
Check this box to indicate that existing help IDs defined in the application will have a .HTM extension appended to them. This is useful when converting applications that use standard help ids to the HTML help standard.

Note:

This check box also makes it possible to have a WinHelp and HTML Help project maintained by the same application. The HTML Help template will strip out the leading tilde (~) from a WinHelp ID.

There is also a new Utility template that will generate all HTML Help Ids to an output text file. See Also the *ListHLPIds Utility Template*.

cwHHProc extension template

The cwHHProc extension template must be put on every procedure that will call the help file. This template allows the help file to be different for each procedure, allows a specific topic page to be opened for the procedure and allows a customized keystroke to call the help file.

Requirements

This control template requires the cwHHGlobal global extension.

Populating the Template

Open the procedure that will contain the HTML Help extension.

1. **Press** the Extensions button.

   The Extensions and Control Templates dialog displays.

2. **Press** the INSERT button and select the cwHHProc extension.

Template Prompts

The cwHHProc template extension provides the following prompts:

Override Help File name
Specifies the name of the procedure's compiled help (.chm) file. If a file is specified for the procedure, it will override the application's default help file. This may be left blank if there is no overriding help file.

Context URL
Specifies a compiled help (.chm) file or a topic within a help file that is to display when the user presses the F1 key. To specify a help topic use the 'SectionName/PageName.htm' naming convention. If the topic page is in the default section you simply use 'PageName.htm'. Remember that when using the Section/Page convention, the slash is a forward slash.

Alternate 'TRAP' Key
Ordinarily the F1 keystroke initiates the call to the help file. This can be changed by entering the equate for the new keystroke to initiate the call to the help file.

AlinkLookUp code template

The AlinkLookUp code template is used to look up one or more Associative link (Alink) names within a compiled help (.chm) file. Associative links are used to link related help topics to each other. When a link that contains an Alink is clicked, a popup window appears with the list of related topics. This template calls the AlinkLookUp method. This method takes three parameters, list of Alinks, default message text and default window title.

Example method calls:

```
oHH.ALinkLookUp( sSearch, sMsg, sTitle ) !Find sSearch Alink
```

Template Prompts:

Alink to search for

Specify the Associative link(s) to search for. A string or variable may be specified. Multiple Alinks can be searched for by entering a list of Alinks separated by semicolons (;) as the string to search for. If a variable is used, the variable should contain a list of Alinks separated by semicolons. This entry is required.

String message to display if not found (opt)

Specifies the message to display in a message box if the Alink is not found. This entry is optional.

Title for Message box (opt)

Specifies the title of the message box dialog. This entry is optional.

CloseHelp code template

The CloseHelp code template closes any Help windows opened by the application. This template calls the CloseHelp method. There are no parameters for this method.

Example method calls:

```
oHH.CloseHelp() !Close all Help windows
```

Template Prompts:

There are no prompts for this code template.

GetHelpFileName code template

The GetHelpFileName code template retrieves the name of the compiled help (.chm) file. This template calls the GetHelpFile method. There are no parameters for this method. This method returns the Help file name.

Example method calls:

```
sHelpFileName = oHH.GetHelpFile( ) !Get Help file name
```

Template Prompts:

Variable to receive the Help File Name

Specifies a variable to be used to receive the HTML Help file name. Use the ellipsis (…) to select the variable from the field lists or type in a variable name that already exists.

GetTopicName code template

The GetTopicName code template gets the current help topic from the help object. This template calls the GetTopic method. This method does not take any parameters but returns the TopicName.

Example method calls:

```
sHelpFileName = oHH.GetTopic( ) !Get topic name
```

Template Prompts:

Variable to receive the Help Topic

Specifies a variable to receive the current help topic. Use the ellipsis (…) to select the variable from the field list or type in a variable name that already exists.

KeyWordLookUp code template

The KeyWordLookUp code template is used to look up one or more Keywords within a compiled help (.chm) file. Keywords are a collection or words and phrases that make up the help file's index. They are used find specific help topics. This template calls the KeywordLookUp method. This method takes three parameters, list of keywords, default message text and default window title.

Example method calls:

```
oHH.KeyWordLookUp( 'Demo' )
```

Template Prompts:

Key word to search for
Specify the Keywords(s) to search for. A string or variable may be specified. Multiple keywords can be searched for by entering a list of Keywords separated by semicolons (;) as the string to search for. If a variable is used, the variable should contain a list of keywords separated by semicolons. This entry is required.

String message to display if not found (opt)
Specifies the message to display in a message box if the Keyword is not found. This entry is optional.

Title for Message box (opt)
Specifies the title of the message box dialog. This entry is optional.

SetHelpFileName code template

The SetHelpFileName code template sets the name of the compiled help (.chm) file that the application will use. This template calls the SetHelpFile method. There is one parameter for this method. The parameter is the help file name.

Example method calls:

```
oHH.SetHelpFile( 'Demo.chm' ) !Set help file name for application
```

Template Prompts:

New HTML Help File Name
Specifies the help file to use for the application. This may be a string or a variable.

SetTopicName code template

The SetTopicName code template sets the current topic name. This should be set before displaying the topic. This template calls the SetTopic method. This method takes one parameter. It is the help topic name.

Example method calls:

```
oHH.SetTopic( 'Class_Interface/Class_Interface.htm' ) !Set topic name
```

Template Prompts:

New Topic

Specifies a topic page within the compiled help (.chm) file. To specify a help topic use the 'SectionName/PageName.htm' naming convention. If the topic page is in the default section you simply use 'PageName.htm'. Remember that when using the Section/Page convention, the slash is a forward slash.

ShowIndex code template

The ShowIndex code template opens the Index tab in the Navigation pane of the HTML Help Viewer and searches for the keyword, if specified. This template calls the ShowIndex method. This method takes one optional parameter, the keyword to search for.

Example method calls:

```
oHH.ShowIndex(  )    !Opens Index tab
oHH.ShowIndex( 'Demo' ) !Opens Index tab and searches for keyword
```

Template Prompts:

Key word to search for (opt)

Specify a keyword to search for in the index. This entry is optional. If no keyword is specified, the Index tab is opened.

ShowSearch code template

The ShowSearch code template opens the Search tab in the Navigation pane of the HTML Help Viewer. This template calls the ShowSearch method. This method has no parameters.

Example method calls:

```
oHH.ShowSearch( )
```

Template Prompts:

There are no prompts for this code template.

ShowTOC code template

The ShowTOC code template selects the Contents tab in the Navigation pane of the HTML Help Viewer. This template calls the ShowTOC method. This method has no parameters.

Example method calls:

```
oHH.ShowSearch( )
```

Template Prompts:

There are no prompts for this code template.

ShowTopic code template

The ShowTopic code template opens the help topic specified by the SetTopicName code template (SetTopic method) in the HTML Help Viewer.

Example method calls:

```
oHH.SetTopic(  )
```

Template Prompts:

There are no prompts for this code template.

Crystal Reports Overview

Seagate Software's Crystal Reports is one of the leading report writers delivering Windows reports. For more information on this product see Seagate Software at www.seagatesoftware.com.

Clarion's Crystal Report interface is comprised of templates, libraries, and DLLs that communicate with Seagate's Crystal Reports, version 8. The DLL is accessed by a Class Interface and is hooked to your application using simple standard Clarion code. This interface allows a seamless integration of previously defined Crystal reports within a Clarion application. The Crystal report engine accesses data and creates the report. The report can be previewed in a Clarion window.

Clarion's Crystal Reports implementation is compatible with both the ABC and Legacy templates. It can only be used in 32-bit applications.

Crystal Reports Files

The files that make up Clarion's Crystal Reports interface are:

C60cr8.inc	Crystal Class Definition
C60cr8L.inc	Crystal Class Definition Local Compile
C60cr8.tpl	Templates
C60cr8.lib	Crystal LIB
C60cr8L.lib	Crystal Local LIB
C60cr8.dll	Crystal DLL

Files to Distribute

To distribute an application that includes the Crystal Report interface, the following files must be distributed.

C60cr8.dll	Crystal DLL

This DLL file must be included in distribution when an application is compiled using the Standalone (C60RUNx.DLL) Run-Time library.

Other Runtime Distribution Files

According to the latest Crystal Reports documentation (Version 9), the required runtime files for an application developed using the Crystal Report Print Engine API (crpe32.dll) are listed below.

The Crystal Reports Print Engine is now considered a legacy API and will no longer expose calls for any the new features included in Crystal Reports. For more information, see "Retired Developer APIs" in *Crystal Reports Developer's Guide (CrystalDevHelp.chm)*.

File	Developer/Distribution Locations Description (Locations are the same unless listed separately.)
crpe32.dll	\Program Files\Common Files\Crystal Decisions\2.0\bin Interface to the Crystal Report Engine
crqe.dll	\Program Files\Common Files\Crystal Decisions\2.0\bin Crystal Query Engine
Implode.dll	\Program Files\Common Files\Crystal Decisions\2.0\bin Compression Library
msvcrt.dll	\Windows\system32 or \WINNT\system32 (NT only) Microsoft (R) C Runtime Library
querybuilder.dll	\Program Files\Common Files\Crystal Decisions\2.0\bin Crystal Query Builder
riched20.dll	\Windows\system32 or \WINNT\system32 (NT only) Rich Text Edit Control, v3.0
ufmanager.dll	\Program Files\Common Files\Crystal Decisions\2.0\bin Manager for loading UFL's
unicows.dll	See the note below. Unicode Layer for Win9x
usp10.dll	\Windows\system32 or \WINNT\system32 (NT only) Uniscribe Unicode script processor

The following DLLs need only be included under these circumstances:

Include Implode.dll only for applications using reports created in versions previous to Crystal Reports 8.

Include unicows.dll only for applications installed on Win9x machines. Due to licensing restrictions, you must get unicows.dll from the Microsoft web site.

Go to http://www.microsoft.com/msdownload/platformsdk/sdkupdate/default.htm? and select "Microsoft Layer for Unicode on Windows 95/98/ME Systems (MSLU) version 1.0".

If your application uses any of the following functions,

CrPEExportToDisk *CrPEExportToExch* *crPEExportToMapi*

CrPEExportToHTML *CrPEExportToODBC* *crPESelectPrinter*

CrPEGetSelectedPrinter

you must include:

crwrap32.dll \Program Files\Common Files\Crystal Decisions\2.0\bin
Print Engine Wrapper (intended as a sample of how to prototype structured calls for languages that do not support structures)

See your Crystal Reports Documentation for any additional runtime requirements.

PreviewCrystalReport

This template allows for previewing and printing of predefined Crystal Reports.

Requirements

There are no requirements for this template.

Populating the Template

1. From an embed point, press the **Insert** button and select the **PreviewCrystalReport** template.

Template Prompts

The PreviewCrystalReport template provides the following prompts:

General

Object Name
Set the object's label for the template-generated code. By fine-tuning the object names, you can make your generated code easier to read. The default object name is oCrystal8.

Report Name or variable (prefixed with !)
Type a valid Crystal report name including extension (.rpt) or a variable that will contain the report name. When using a variable, precede it with an exclamation (!). The report name must include the runtime path of the file. If the file is not found at runtime an error window will display containing the following error- "error opening job".

Window Title or variable (prefixed with !)
Crystal Reports are run inside a Clarion window. Type a title to appear in the window caption bar or enter a variable name which will contain the window title. When using a variable, precede it with an exclamation (!).

Show Print Controls?
Check this box to turn off all preview window buttons. If checked, the **Control Options** template tab will not be available.

Control Options

This tab will not show if the **Show Print Controls?** option on the **General** tab is not checked.

Allow Prompting?
Check this box to allow the Crystal Report use its defined runtime parameter fields.

Allow Drill Down?
Check this box to allow support of Crystal's drill-down report feature.

Show Cancel Button?
Check this box to enable a **Cancel** button on the report preview window.

Show Close Button?
Check this box to enable a **Close** button on the report preview window.

Show Export Button?
Check this box to enable an **Export** button on the report preview window.

Show Launch Button?
Check this box to enable a **Launch** button on the report preview window. The Launch button runs the Seagate Analysis tool.

Show Navigation Buttons?
Check this box to enable page navigation buttons on the report preview window.

Show Print Button?
Check this box to enable a **Print** button on the report preview window.

Show Print Setup?
Check this box to enable a **PrintSetup** button on the report preview window.

Show Progress?
Check this box to enable the report progress controls in the preview window. These controls include Total Records, Records Read, Percent Read.

Show Refresh?
Check this box to enable a **Refresh** button on the report preview window.

Show Search?
Check this box to enable the two search controls on the report preview window. These two controls include a search button and an entry control.

Show Zoom?
Check this box to enable the zoom control on the report preview window.

Show Toolbar Tips?
Check this box to enable tool tips on the toolbar in the preview window.

Show Document Tips?
Check this box to enable tool tips on the document (report) in the preview window.

Window Options

Initial State

Select the initial size and state of your window from the drop-down list. Choose from:

Normal Display the window at the default size. If you don't specify a default size, Clarion's run-time library sets it for you.

Maximized This window fills the entire desktop, or the entire application frame, depending on whether the window is an application window, or an MDI child window.

Iconized In Windows 3.1, the window appears in an iconized state–as a 32 by 32 pixel window at the bottom of the desktop (application window) or at the inside bottom of the application frame (MDI child window).

In Windows 95, the window appears in an iconized state in the Taskbar.

Frame Type

Pick the frame type for your window from the drop-down list. Choose from:

Single A single pixel frame that 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 (without a caption bar), or for a modal dialog box (with a caption bar).

Resizable A thick frame, which the user *may* resize. Choose this for application and MDI child windows.

None A single pixel frame under Windows 95/98, and no frame under Windows 3.1. Most suitable for dialog boxes. The user cannot resize this frame.

Icon

To associate an icon with the window (and add a system menu—see **System Menu** below), specify an icon file name (.ICO file) in this field. Type the file name or press the ellipsis button (...) to select a file name with the standard **Open File** dialog. Specifying an icon automatically places a minimize button on the caption bar of your application or MDI child window.

System Menu

To place a system menu in your window, check the **System Menu** box, or specify an icon file (see **Icon** above), or specify a maximize box (see **Maximize** below). When your window has the SYSTEM attribute Windows 95/98 and Windows NT display an icon in the upper left corner. If you specify an icon (see **Icon** above), that icon displays, otherwise the system default icon displays. Initially, the system default icon is set to the Clarion icon; however, you can specify a system default icon with:

```
System{PROP:Icon} = 'My.ico'
```

Activate the system menu by CLICKING the button, box, or icon in the upper left corner of the window. Standard system menu choices include Restore, Minimize, Maximize, and Close.

Every application frame 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 application window.

Maximize Box

To place a maximize button in your window (and a system menu—see **System Menu** above), check this box. In general, you should place a maximize button on application windows and MDI child document windows, not on dialog boxes.

3D Look

To provide the gray window background, and chiseled control look for your window, check this box. This is clearly a style consideration, but will go a long way in giving your application a professional look.

PrintCrystalReport

This template allows for printing of predefined Crystal Reports. The report cannot be previewed with this template. An optional printer setup dialog is available.

Requirements

There are no requirements for this template.

Populating the Template

1. From an embed point, press the **Insert** button and select the **PrintCrystalReport** template.

Template Prompts

The PrintCrystalReport template provides the following prompts:

Object Name
Set the object's label for the template-generated code. By fine-tuning the object names, you can make your generated code easier to read. The default object name is oCrystal8.

Report Name or variable (prefixed with !)
Type a valid Crystal report name including extension (.rpt) or a variable that will contain the report name. When using a variable, precede it with an exclamation (!). The report name must include the runtime path of the file. If the file is not found at runtime an error window will display containing the following error- "error opening job".

Number of Copies or variable
Specify the number of copies to print as a hard coded numeric value or specify a variable (prefixed with !) to set the number of copies at runtime.

Show Printer Setup?
Check this box to show the printer setup dialog before the report is sent to the printer. This allows the user to select a specific printer and set printer properties.

GetCrystalFormulaPreview
GetCrystalFormulaPrint

The **GetCrystalFormulaPreview** and **GetCrystalFormulaPrint** templates both accomplish the exact same task. They retrieve the Formula that is used to limit retrieved records. The difference between these two templates is that **GetCrystalFormulaPreview** is used in conjunction with the **PreviewCrystalReport** template and **GetCrystalFormulaPrint** is used in conjunction with the **PrintCrystalReport** template.

These templates call the Crystal8.SelectionFormula method.

Requirements

These templates require the **PreviewCrystalReport** or the **PrintCrystalReport** template respectively.

Populating the Template

1. From an embed point, press the **Insert** button and select either the **GetCrystalFormulaPreview** or **GetCrystalFormulaPrint** template.

Template Prompts

These templates provides the following prompts:

Target Variable

Specify a valid variable to store the retrieved Formula. This may be a STRING, CSTRING, PSTRING, or MEMO data type. The variable may be a local, module, or global variable. You may also use a file field; however, you must write the code to update the file. This variable is required.

GetCrystalQueryPreview GetCrystalQueryPrint

The **GetCrystalQueryPreview** and **GetCrystalQueryPrint** templates both accomplish the exact same task. They retrieve the SQL Query that is sent to the SQL data source. The difference between these two templates is that **GetCrystalQueryPreview** is used in conjunction with the **PreviewCrystalReport** template and **GetCrystalQueryPrint** is used in conjunction with the **PrintCrystalReport** template.

These templates call the Crystal8.Query method. For more information on these methods refer to the *ABC Library Reference, Crystal8 class.*

Requirements

There templates require the **PreviewCrystalReport** or the **PrintCrystalReport** template respectively.

Populating the Template

1. From an embed point, p**ress** the **Insert** button and select either the **GetCrystalQueryPreview** or **GetCrystalQueryPrint** template.

Template Prompts

These templates provides the following prompts:

Target Variable

Specify a valid variable to store the retrieved SQL query. This may be a STRING, CSTRING, PSTRING, or MEMO data type. The variable may be a local, module, or global variable. You may also use a file field; however, you must write the code to update the file. This variable is required.

SetCrystalFormulaPreview SetCrystalFormulaPrint

The **SetCrystalFormulaPreview** and **SetCrystalFormulaPrint** templates both accomplish the exact same task. They set the Formula that is used to limit retrieved records. The difference between these two templates is that **SetCrystalFormulaPreview** is used in conjunction with the **PreviewCrystalReport** template and **SetCrystalFormulaPrint** is used in conjunction with the **PrintCrystalReport** template.

These templates call the Crystal8.SelectionFormula method. For more information on these methods refer to the *ABC Library Reference, Crystal8 class.*

Requirements

There templates require the **PreviewCrystalReport** or the **PrintCrystalReport** template respectively.

Populating the Template

1. From an embed point, p**ress** the **Insert** button and select either the **SetCrystalFormulaPreview** or **SetCrystalFormulaPrint** template.

Template Prompts

These templates provides the following prompts:

Formula Variable

Specify a valid variable that holds the Formula to send to the report. This variable may be a STRING, CSTRING, PSTRING, or MEMO data type. The variable may be a local, module, or global variable. You may also use a file field. This variable is required.

SetCrystalQueryPreview
SetCrystalQueryPrint

The **SetCrystalQueryPreview** and SetCrystalQueryPrint templates both accomplish the exact same task. They set the SQL Query that wiil be sent to the SQL data source. The difference between these two templates is that **SetCrystalQueryPreview** is used in conjunction with the **PreviewCrystalReport** template and **SetCrystalQueryPrint** is used in conjunction with the **PrintCrystalReport** template.

These templates call the Crystal8.Query method. For more information on these methods refer to the *ABC Library Reference, Crystal8 class.*

Requirements

These templates require the **PreviewCrystalReport** or the **PrintCrystalReport** template respectively.

Populating the Template

1. From an embed point, p**ress** the **Insert** button and select either the **SetCrystalQueryPreview** or **SetCrystalQueryPrint** template.

Template Prompts

These templates provides the following prompts:

Query Variable
Specify a valid variable that holds the SQL Query to send to the SQL data source. This may be a STRING, CSTRING, PSTRING, or MEMO data type. The variable may be a local, module, or global variable. You may also use a file field. This variable is required.

E-commerce Templates

Overview

The e-commerce template sets are used to connect with online credit card processing services. These services include a connection to Cybercash Inc. and Linkpoint International. Both services work with a financial processor to get payments authorized and credited to the involved parties.

The e-commerce templates include the following:

CyberCashSettings	CyberCash Global Settings
CapturePayment	Process CyberCash Transactions
LinkPointGlobal	Linkpoint Global Settings
ProcessOrder	Process Linkpoint Transactions

Template Components

Clarion's CyberCash and Linkpoint support each have their own template classes. Class CyberCash is defined in ABCCASH.TPL. Class Linkpoint is defined in ABLINKP.TPL.

CyberCash Templates

CyberCash was a world leader in e-commerce technologies and services, enabling commerce solutions for physical store retailers and the Internet. VeriSign, the leading provider of digital trust services, recently acquired CyberCash's Internet payments business.

CyberCash provided a complete line of software products and services allowing merchants, billers, financial institutions and consumers to conduct secure transactions and other e-commerce functions using a broad array of popular payment forms. Credit, debit, purchase cards, cash, checks, smart cards and alternative payment types (e.g. "frequent buyer" or loyalty programs) are all supported by CyberCash payment solutions. For more information, CyberCash Inc. can be found at www.cybercash.com. This link will redirect you to the VeriSign web site.

To use the CyberCash templates you must first setup a merchant account with CyberCash as well as a relationship with a financial processor that supports CyberCash. The financial processor is responsible for deducting money from the customer's credit card and transferring it to the merchant bank account.

CyberCashSettings Template

The CyberCash global extension template is used to setup and maintain payment services over the Internet. The CyberCashSettings template is required in order for the application to use Clarion's CyberCash support. This extension simply includes the necessary equates and Classes needed to interface to CyberCash.

It uses CURRENCY.INC for country specific currency codes.

This extension template does not rely on any other templates.

Settings

The list box shows all accounts that have been setup to interface with CyberCash. Press the **Insert**, **Properties**, or **Delete** buttons to modify the account settings.

CyberCash Account Setup

Auto Mark? Check this box to turn on the auto-mark function for transactions. This function automatically marks your transactions for inclusion in a batch.

Auto Settle? Check this box to turn on the auto-settle function for transaction batches. The auto-settle function automatically initiates the capture of a batch at the end of the day.

Emulate HostMode?
Check this box to allow payment authorization and settlement to occur in a single step. This type of setup is normally used for monthly service fee situations or when the purchase is made available for immediate use. If this option is not used, the payment is first authorized. It must then be submitted for monetary settlement.

Currency Select the currency region from the drop down list box. This sets all CyberCash transactions to be done using the currency code for the selected currency region. The regions are defined in CURRENCY.INC. This file is defined based on the ISO 4217, 1st revision 1981 and defines all currency regions and their associated currency code.

Processor URL
Specifies the URL of the payment processor's host. For CyberCash the default is http://cr.cybercash.com/cgi-bin/.

Proxy Host Specify the hostname of the computer that is used as the HTTP proxy server. This is the computer that will receive information through the proxy server. This field is optional.

Proxy Port Specify the port number that the computer will use to receive information through the proxy server. This field is enabled and must be used when a Proxy Host is specified

Timeout Specify the number of seconds to allow for communications before the system stops communicating.

Login (CYBERCASH_ID)
A unique alphanumeric identifier assigned to you (the merchant) by CyberCash during the initial merchant registration process.

Password Enter the password (hash secret value). This is provided by the CyberCash registration system.

Key (MERCHANT_KEY)
Enter your merchant key. This is provided by the CyberCash registration system.

Classes

The Classes tab lets you control the classes (and objects) the procedure uses. You may accept the default Application Builder Class and its object (recommended), or you may specify your own or a third party class. 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.

CapturePayment

The CapturePayment extension template provides the ability to process online credit card based transactions. Transactions are first authorized with the merchant service provider. Once a transaction is authorized it is submitted for settlement to the service provider for payment. The service provider and the selected financial processor arrange for the credit card to be charged and the money transferred ending up in the merchant account.

Event Settings

Authorize Authorization occurs for all credit card transactions. This process submits the purchase request to the credit card bank. If the card has credit available, the transaction is approved.

Control Select a control from the drop down list. When this control is accepted at runtime, the authorization process occurs.

Capture Capture (settlement) is the process where a previously authorized transaction is posted for monetary settlement. This is normally done when purchase is delivered to the customer. At this time, funds are transferred into the merchant's account. This control is disabled and not used if the Emulate HostMode? option is set because the capture is handled automatically when the transaction is authorized.

Control Select a control from the drop down list. When this control is accepted at runtime, the capture process occurs.

Submit Submit allows previously authorized transactions to be manually submitted for settlement. This option should not be used if the global **Emulate HostMode?** option is set.

Control Select a control from the drop down list. When this control is accepted at runtime, the submission process occurs.

Return Return allows a merchant to post a transaction that will transfer money from the merchant's account to the customer's credit card as a credit.

Control Select a control from the drop down list. When this control is accepted at runtime, the submission of a return occurs.

Card Settings

Currency Specifies the column from a table that designates the currency code to be used for the transactions. CyberCash limits one currency type per account. If more than one currency is to be used, multiple merchant accounts must be setup with CyberCash.

Card Number Specifies the column from a table that holds the customers credit card number. Use the ellipsis (…) to select the column from the table or type in a column name that exists in the table.

Expiry Date Specifies the column from a table that holds the customers credit card expiration date. Use the ellipsis (…) to select the column from the table or type in a column name that exists in the table.

Card Name Specifies the column from a table that holds the credit card owner's name. Use the ellipsis (…) to select the column from the table or type in a column name that exists in the table.

Card Address Specifies the column from a table that holds the credit card owner's address. Use the ellipsis (…) to select the column from the table or type in a column name that exists in the table.

Card City Specifies the column from a table that holds the credit card owner's city. Use the ellipsis (…) to select the column from the table or type in a column name that exists in the table.

Card Zip Specifies the column from a table that holds the credit card owner's zip code. Use the ellipsis (…) to select the column from the table or type in a column name that exists in the table.

Card State Specifies the column from a table that holds the credit card owner's state. Use the ellipsis (…) to select the column from the table or type in a column name that exists in the table.

Card Country Specifies the column from a table that holds the credit card owner's country. Use the ellipsis (…) to select the column from the table or type in a column name that exists in the table.

Authorization Settings

Transaction Name

Specify a string that identifies transactions. This may be a string up to 255 characters. The transaction name is concatenated with an invoice number to produce a unique transaction id.

Authorization Amount

Specifies the column from a table that holds the amount to be authorized. Use the ellipsis (…) to select the column from the table or type in a column name that exists in the table.

Invoice Number

Specifies the column from a table that holds the invoice number. Use the ellipsis (…) to select the column from the table or type in a column name that exists in the table.

Capture Amount

Specifies the column from a table that holds each transactions settlement amount. The total of these transactions must equal the Captured Total. Use the ellipsis (…) to select the column from the table or type in a column name that exists in the table.

Captured Total

Specifies the column from a table that holds the total settlement amount. Use the ellipsis (…) to select the column from the table or type in a column name that exists in the table.

Authorization Field

Specifies the column from a table which receives the authorization code. Use the ellipsis (…) to select the column from the table or type in a column name that exists in the table.

Processing Error Field

Specifies the column from a table that holds the error field. If an error occurs in processing, this field will contain the error message from CyberCash. Use the ellipsis (…) to select the column from the table or type in a column name that exists in the table.

Date Billed Field

Specifies the column from a table that holds the billing date. This value of this field is supplied if this template is used within a PROCESS procedure type and the transaction is authorized and captured and settled. Use the ellipsis (…) to select the column from the table or type in a column name that exists in the table.

Debug LogFile

Specify the name of a file to use to capture all data that is sent to and received from CyberCash. Use the ellipsis (…) to select a path and/or file from the FileDialog or type in a filename.

LinkPoint Templates

Linkpoint International provides a Secure Payment Gateway solution that allows customers to purchase products online. The service communicates with the credit card processor to authorize credit card transactions in real-time. Included with this Linkpoint support is tax and shipping calculations and electronic software download (ESD). For more information, Linkpoint International can be found at www.linkpoint.com.

LinkPointGlobal

The LinkPointGlobal extension template provides the ability to access the Secure Payment Gateway engine to process credit card transactions in real-time. This template is used to setup the communication channel and is required in order to use the ProcessOrder extension template.

Requirements

This extension template has no other template requirements.

Populating the Template

1. Press the Global icon button from the IDE.
2. Press the **Extensions** button.
3. Press the **Insert** button and select the *LinkPointGlobal* extension.

Template Prompts

The LinkPointGlobal template provides the following prompts:

LinkPoint Settings

Config File
Specifies a string, variable, or runtime expression using EVALUATE to use as the configuration file name which stores the information needed to connect to the LinkPoint engine. The filename should be the same as the merchant configuration file name without the .config extension. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=). This field is required.

Host
Specifies a string, variable, or runtime expression using EVALUATE to use as the name of the server that LinkPoint is running on. The value contained in this field may be a fully qualified hostname or an IP address. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=). This field is required.

Port

Specifies the configured communication port for Linkpoint. This field is required. The engine can be configured to run on any port, however the default port is 1139.

Key

Specifies a string, variable, or runtime expression using EVALUATE to use as the path to the SSL merchant certificate. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=). This field is required.

Mode

Choose a runtime mode from the drop down list box.

Test
Choose Test to run some sample transactions. The credit card number 4111111111111111 can be used for testing.

Live
Change the mode to Live after all testing is complete.

Classes

The Classes tab lets you control the classes (and objects) the procedure uses. You may accept the default Application Builder Class and its object (recommended), or you may specify your own or a third party class. 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.

ProcessOrder Extension Template

The ProcessOrder extension template works in conjunction with the LinkPointGlobal extension. It uses a PROCESS procedure type to process orders. If the order total is greater than 0, the order is processed and the credit card is charged.

The file schematic for the process procedure should have the invoice header file as the primary file. The related customer and invoice detail files should be available in the file schematic. As the invoice header file is read, the related customer and item detail files are read.

Requirements

This extension template requires the LinkPointGlobal extension template and can only be used on a PROCESS procedure type.

Populating the Template

1. RIGHT-CLICK on the procedure and choose **Extensions** from the popup menu.
2. Press the **Insert** button and select the *ProcessOrder* extension.

Template Prompts

The ProcessOrder template provides the following prompts:

Override Global Settings
To override the global ClearLink settings (set in the LinkPointGlobal extension) press the Override Global Settings button.

Config File
Specifies a string, variable, or runtime expression using EVALUATE to use as the configuration file name which stores the information needed to connect to the LinkPoint engine. The filename should be the same as the merchant configuration file name without the .config extension. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=). This field is required.

Host
Specifies a string, variable, or runtime expression using EVALUATE to use as the name of the server that LinkPoint is running on. The value contained in this field may be a fully qualified hostname or an IP address. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=). This field is required.

Port
Specifies the configured communication port for Linkpoint. This field is required. The engine can be configured to run on any port, however the default port is 1139.

Key
Specifies a string, variable, or runtime expression using EVALUATE to use as the path to the SSL merchant certificate. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=). This field is required.

Order Settings Tab

The Order Settings tab consists of entries used to identify the necessary settings used to process a customer order.

IP Address
Specifies the column from a table that holds the customer's IP address. This data is used in conjunction with the Fraud Protection Module. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a CSTRING data type with a maximum of 20 characters.

Unique Order ID
Specifies the column from a table that holds a unique order identifier. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a CSTRING data type with a maximum of 100 characters and is required by the LinkPoint engine.

Charge Type

Choose a valid charge type from the drop-down list box. The default charge type is Chargetype_Sale.

Chargetype_Sale
This charge type charges credit card immediately. This should be used when products are transferred electronically. For other purchases funds must be pre-authorized on the credit card until the product(s) is shipped.

Chargetype_Preauth
This charge type reserves the funds on a credit card. This should be used when selling goods that must be physically shipped. Credit card networks reserve the funds for varying amounts of time, ranging from three days to several months.

Chargetype_Postauth
This charge type charges the credit card based on the funds that were reserved with the charge type Chargetype_Preauth.

Chargetype_Credit
This charge type is used to refund a customer's credit card from the merchant's account.

UserID
Specifies the column from a table that holds the unique customer id. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a CSTRING data type with a maximum of 254 characters and is required by the LinkPoint engine.

Cardnumber
Specifies the column from a table that holds the credit card number of the purchaser. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a CSTRING data type with a maximum of 24 characters and is required by the LinkPoint engine. Spaces and dashes included in the data are ignored by the LinkPoint engine.

Expiry month
Specifies the column from a table that holds the two-digit expiration month number of the customers credit card. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a CSTRING data type with a maximum of 3 characters and is required by the LinkPoint engine.

Expiry year
Specifies the column from a table that holds the two-digit expiration year number of the customers credit card. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a CSTRING data type with a maximum of 3 characters and is required by the LinkPoint engine. The format of the data contained in this field must be YY.

Address Number
Specifies the column from a table that holds the numeric part of the bill to street address of the customer. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a CSTRING data type with a maximum of 10 characters.

eMail Address
Specifies the column from a table that holds the customer's email address. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a CSTRING data type with a maximum of 80 characters. If no email address is supplied no email confirmation will be sent to the customer.

Order Total
Specifies the column from a table that holds the total order cost. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as an SREAL data type and is required by the LinkPoint engine.

Phone
Specifies the column from a table that holds the customer's phone number. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a CSTRING data type with a maximum of 254 characters.

Fax
Specifies the column from a table that holds the customer's fax number. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a CSTRING data type with a maximum of 254 characters.

Comments
Specifies the column from a table that holds extra comments regarding the order. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a CSTRING data type with a maximum of 254 characters.

Billing Details

The Billing Details tab consists of entries used to identify the bill to name and address columns from the customer table.

Name
Specifies the column from a Customer table that holds the bill to name of the customer. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a CSTRING data type with a maximum of 254 characters.

Company
Specifies the column from a Customer table that holds the bill to company name of the customer. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a CSTRING data type with a maximum of 254 characters.

Address
Specifies the column from a Customer table that holds the bill to street address of the customer. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a CSTRING data type with a maximum of 254 characters.

City
Specifies the column from a Customer table that holds the bill to city of the customer. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a CSTRING data type with a maximum of 254 characters.

State
Specifies the column from a Customer table that holds the bill to state of the customer. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a CSTRING data type with a maximum of 254 characters.

Zip
Specifies the column from a Customer table that holds the bill to zip code of the customer. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a CSTRING data type with a maximum of 254 characters.

Country
Specifies the column from a Customer table that holds the bill to country of the customer. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a CSTRING data type with a maximum of 254 characters.

Item Details

The Item Details tab consists of entries used to identify the columns from the proper tables used as billing information.

Item ID
Specifies the column from an Invoice Detail table that holds the product id of the item being purchased. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a CSTRING data type with a maximum of 254 characters.

Description
Specifies the column from an Invoice Detail table that holds the description of the item being purchased. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a CSTRING data type with a maximum of 254 characters.

Price
Specifies the column from an Invoice Detail table that holds the price of the item being purchased. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as an SREAL data type.

Quantity

Specifies the column from an Invoice Detail table that holds the quantity of the item being purchased. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a LONG data type.

Tax Details

The Tax Details tab consists of entries used to identify the columns from the proper tables used to calculate U.S. sales tax.

State
Specifies the column from a table that holds the taxing state. The state column should contain a two-digit state code in accordance with the state codes used by the ClearLink engine. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. This column should be defined as a CSTRING data type with a maximum of 3 characters.

Zip
Specifies the column from a table that holds the 5 character-taxing zip codes. Use the ellipsis (…) to select the field from the file or type in a field that exists in the file. The column should be defined as a CSTRING data type with a maximum of 6 characters.

E-mail Related Templates

The e-mail templates are used to send email (SMTP) or newsgroup (NNTP) messages from a Clarion application. Messages can be manually sent to one or more recipients and can be triggered on certain events or actions within the program. These templates include addressbook support and messages attachments. Plain text or HTML messages are supported.

The e-commerce templates include the following:

GlobalDocumentHandling	Messaging Setup
DocumentSend	Send Messages
LoadAddresses	Load AddressBook
FileTriggers	Trigger Messages

Template Components

Clarion's e-mail templates have a template class of Messaging. This class of templates is defined in ABMAIL.TPL.

GlobalDocumentHandling

The GlobalDocumentHandling template adds email messaging support to your application. It handles the global messaging settings such as SMTP and global addresses. It is a global extension which the DocumentSend, LoadAddresses, and FileTriggers extension templates require in order to send messages.

SMTP Settings

Sender address Specifies a string, variable, or runtime expression using EVALUATE which contains a valid email address indicating who is transmitting the message. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=). This entry is required.

From Header address Specifies a string, variable, or runtime expression using EVALUATE which contains a valid email address indicating who is sending the message. This will show in the message header. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

From Header Name Specifies a string, variable, or runtime expression using EVALUATE which contains the name of the user who is sending the message. This will show in the message header. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Sender Domain Specifies a string, variable, or runtime expression using EVALUATE which contains the host domain name. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Default SMTP Server Specifies a string, variable, or runtime expression using EVALUATE which contains the default SMTP server to use if there is none specified when establishing a connection. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Default SMTP Port Specifies a string, variable, or runtime expression using EVALUATE which contains default SMTP port to use. The default value is set to 25. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Reply To List Specifies one or more mailboxes to which replies to an email are sent.

Reply To Address Specifies a string, variable, or runtime expression using EVALUATE which contains an email address to which replies to an email are sent. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Reply To Name Specifies a string, variable, or runtime expression using EVALUATE which contains a users name to which replies to an email are sent. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

User Defined Headers Specifies one or more user defined mail headers. Each user defined header is prepended with an 'X-' which is required by the SMTP specifications.

Header Specifies a string, variable, or runtime expression using EVALUATE which contains a user defined header field. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Value Specifies a string, variable, or runtime expression using EVALUATE which contains the user defined header field value. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

NNTP Settings

From Header address Specifies a string, variable, or runtime expression using EVALUATE which contains a valid email address indicating who is sending the message. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

From Header Name Specifies a string, variable, or runtime expression using EVALUATE which contains the name of the user who is sending the message. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Default NNTP Server Specifies a string, variable, or runtime expression using EVALUATE which contains the default NNTP server to use if there is none specified when establishing a connection. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Default NNTP Port Specifies a string, variable, or runtime expression using EVALUATE which contains default NNTP port to use. The default value is set to 119. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Username Specifies a string, variable, or runtime expression using EVALUATE which contains a user name for News servers which require authorization. The default value is set to 119. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Password Specifies a string, variable, or runtime expression using EVALUATE which contains a password for News servers which require authorization. The default value is set to 119. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

User Defined Headers Specifies one or more user defined mail headers. Each user defined header is prepended with an 'X-' which is required by the SMTP specifications.

Header Specifies a string, variable, or runtime expression using EVALUATE which contains a user defined header field. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Value Specifies a string, variable, or runtime expression using EVALUATE which contains the user defined header field value. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Global Address Settings

The Global Address Settings tab lets you initialize the contents of the global address book. The recipients information can be stored in an INI file or in a data file. If they are stored in an INI file, the section name in the INI file is [Addresses]. The entries in this section which begin with 'Global' are email recipients defined for the global address book.

Procedure to load Select a procedure from the drop down list. This procedure will load a group of email recipients into the global address book from a data file. Typically this procedure will be a *Process* procedure that incorporates the LoadAddresses template extension.

Save and Restore Check this box to save and restore global address book entries to an INI file. The INI file specified by the application is used. If the application is not using an INI file, one is created using <programname>.INI.

Other Template Support Info

The Other Template Support Info tab displays template extensions related to messaging and the status of them within the application. This tab is for informational purposes only and should be used to determine if support for a particular option is enabled for the application.

DbAuditing Support?
This is checked when the global DbAuditing extension template is added to the application.

File User Tags Support?
This is checked when the global FileUserTags extension template is added to the application.

Generic Skeletons Support?
This is checked when the global GenericSkeletonClasses extension is added to the application.

Document Merging Support?
This is checked when the global FileUserTags and GenericSkeletonClasses extensions are both added to the application.

Web Guard Support Included?
This is checked when the global Web and WebGuard extensions are added to the application.

Web Guard Support

The Web Guard Support tab is available when the global Web and WebGuard extensions are added to the application. The WebGuard options on this tab work in conjunction with the Global Document Handler classes to support the transmission of emails on invalid logins determined by the WebGuard template.

eMail Reminder on Invalid Login Password?
Check this box to enable emails to be sent to the user if an incorrect password is tried.

Subject Specifies a string, variable, or runtime expression using EVALUATE which contains the subject of the email to be sent to the user. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Message Body Press the ellipsis (…) button to choose a file that contains the body of the message that will be sent to the user. .htm or .txt files are available to select from. You can also type the variable name directly into the entry box.

Content Type Select a content type based on the type of file specified for the message body. If a .htm file was selected for the message body, choose *Html* for the content type. If a .txt file was selected for the message body, choose *Plain Text*. The default value is *Html*.

Recipients Specifies the destination of the email reminder.

Address Specifies a string, variable, or runtime expression using EVALUATE which contains an email address to which the email is sent. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Name Specifies a string, variable, or runtime expression using EVALUATE which contains a users name to which the email is sent. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Recipient Class Choose from the drop down list which type of recipient you are specifying. Choose **Send To** for a primary recipient. Choose **CC** for a secondary recipient. Choose **BCC** for a hidden recipient.

Email Administrator on Invalid Logins?
Check this box to enable emails to be sent to the system administrator if an incorrect login occurs.

Subject Specifies the subject of the email to be sent to the administrator.

Message Body Press the ellipsis (…) button to choose a file that contains the body of the message that will be sent to the user. .htm or .txt files are available to select from. You can also type the variable name directly into the entry box.

Content Type Select a content type based on the type of file specified for the message body. If a .htm file was selected for the message body, choose *Html* for the content type. If a .txt file was selected for the message body, choose *Plain Text*. The default value is *Html*.

Administrator eMail Details
Specifies the Administrator email information, including email address, name and type of recipient.

Address Specifies a string, variable, or runtime expression using EVALUATE which contains an email address to which the email is sent. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Name Specifies a string, variable, or runtime expression using EVALUATE which contains a users name to which the email is sent. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Recipient Class Choose from the drop down list which type of recipient you are specifying. Choose Send To for a primary recipient. Choose *CC* for a secondary recipient. Choose *BCC* for a hidden recipient.

Classes

The classes tab lets you control the class (and object) the template uses. You may accept the default Application Builder Class and it's object (recommended) or you may specify your own or a third party class. 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.

DocumentSend

The DocumentSend template is used to provide messaging support. Messages can be triggered from a window event or a control event. This template provides address book support to select recipients to receive a message. Messages may be sent using SMTP or NNTP protocols. The template requires the presence of the GlobalDocumentHandling template.

Tip

Adding debug=>1 to the defines of the project you will see the line-by-line communication between the mail client and the mail server.

Controls

Send Message Settings

Send on Control Event

Specifies a message is sent when the specified control's event occurs. There may be one or more control events that trigger a message.

Control Specifies the control that triggers the message. Choose a control from the drop down list of controls which contains all controls populated on the window.

Event Specifies the control specific event to trigger a message. Choose an event from the drop down list of events for the chosen control.

Send On Window Event?

Check this box to send a message when the specified window event occurs.

Send on Window Event

Choose a window event from the drop down list of events.

Send on Method Call?

Check this box to send a message on a method call.

Send on Method Choose a method from the drop down list. Valid method calls are *Window Completed*, *Process a Record*, and *After IReport Creation*.

Address Book Settings

Recipient Listbox

Specifies the list box control that receives the selected message recipients. Select the control from the drop down list box.

Use Global Addresses

Check this box to use the global address book to choose recipients from.

Save and Restore

Check this box to save and restore recipients to the global address book.

Family

Specifies the family entry tied to the local address book. This is the name of the procedure by default.

Address Button

Address Book Trigger

Select a trigger from the drop down list. This trigger is used to open the address book. Valid triggers are *Control Event*, *Window Event*, *Other*, and *None*.

Call Address Book on Control Event

Specifies which control triggers the address book. This is available when the **Address Book Trigger** is set to Control Event.

Control Specifies the control that triggers the address book. Select the control from the drop down list box.

Event Specifies the control event that triggers the address book. Select the event from the drop down list box.

Address Book Window Event

Select a Window Event from the drop down list box to trigger the address book. This is available when the **Address Book Trigger** is set to Window Event.

Address Book on Method

Select a method from the drop down list box to trigger the address book. This is available when the **Address Book Trigger** is set to *Other*. Valid methods are *Window Completed* and *Process a Record*.

Load Addresses

Select a load method from the drop down list box. This is used to load addresses into the address book. Valid load methods are *Once* and *Once per valid record*. This is available only on a Process procedure.

Message Settings

Subject

Specifies a string, variable, or runtime expression using EVALUATE which contains the subject to use in the message. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Priority

Specifies a priority level of an email message.

Priority

Select a priority level from the drop down list box. Valid priority levels are *None*, *High*, *Normal*, and *Low*.

Priority Value

Specify a user defined priority level. This may be a string, variable, or runtime expression using EVALUATE which contains the priority to use in the message. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=). This is only available if the **Priority** is set to *None*. Setting the priority value to High gives the message the highest priority. Setting the priority value to Low gives the message the lowest priority.

Message Body Source

Specifies the body content of the message. The body can be information inserted from the contents of a file or can be the contents of a column in a table. Choose *Field* or *File* from the drop down list.

File

Choose a file to use in the message body by pressing the ellipsis (…) button. This option is available when the **Message Body Source** specifies *File*.

File is a skeleton

Check this box to merge the chosen file with a skeleton html file (skeletons are primarily used for Web Builder applications) to provide personalized information in the body text of the message.

Field Choose a field to use in the message body by pressing the ellipsis (…) button. This option is available when the **Message Body Source** specifies *Field*.

Content Type Specifies the type of message to be sent. Select either *Html* or *Plain Text* from the drop down list.

Encoder Specifies the type of encoding to use. Select either *Quoted Printable* or *Base64* from the drop down list box.

Alternative Html Formatted Message Body
Specify the alternative Html settings for the message body.

Message Body Source
Specifies the body content of the message. The body can be information inserted from the contents of a file or can be the contents of a column in a table. Choose None, *Field* or *File* from the drop down list.

File Choose a file to use in the message body by pressing the ellipsis (…) button. This option is available when the **Message Body Source** specifies *File*.

Field Choose a field to use in the message body by pressing the ellipsis (…) button. This option is available when the **Message Body Source** specifies *Field*.

Encoder Specifies the type of encoding to use. Select either *Quoted Printable* or *Base64* from the drop down list box.

Attachments

Specify one or more attachments to a message.

Attachment Source Select *File* or *Field* from the drop down list box to use as the attachment to a message.

File Choose a file to use as an attachment by pressing the ellipsis (…) button.

Field Choose a field to use as an attachment by pressing the ellipsis (…) button.

Filename Specify the filename of the attachment. Choose the file by pressing the ellipsis (…) button.

Content Type Specifies the type of message to be sent. Select *Html, Plain Text*, *Binary*, or *Other* from the drop down list box.

Other Content Type Specifies the specialized content type of the message. This is used for types such as audio, video, etc. This is available when the **Content Type** is set to *Other*.

Content Sub Type Specifies the content sub type of the message. This is used for sub types such as *octetstream* and *msword*. This is available when the **Content Type** is set to *Other*.

Encoder Specifies the type of encoding to use. Select either *Quoted Printable* or *Base64* from the drop down list box.

Embedded Check this box to make the attachment part of the message text. If not checked, the attachment should be a separate part of the email message.

Static Addressing

Address Settings Setup one or more recipients to receive the message being sent. These are static recipients and will always be included on any message.

Recipient Class Choose from the drop down list which type of recipient you are specifying. Choose *Send To* for a primary recipient. Choose *CC* for a secondary recipient. Choose *BCC* for a hidden recipient.

Recipient Address Specifies a string, variable, or runtime expression using EVALUATE which contains an email address to which the message is sent. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Recipient Name Specifies a string, variable, or runtime expression using EVALUATE which contains a users name to which the message is sent. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Transport to use Specifies the transport protocol to use when sending a message to this recipient. Select a protocol from the drop down list box. Valid transport protocols are SMTP (Simple Mail Transport Protocol) and NNTP (Network News Transport Protocol). The default value is SMTP.

Classes

The classes tab lets you control the class (and object) the template uses. You may accept the default Application Builder Class and it's object (recommended) or you may specify your own or a third party class. 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.

LoadAddresses

The LoadAddresses template is used on a Process procedure to load email addresses from a table into the program's address book. It requires the GlobalDocumentHandling template.

Address Field Specifies a column from a data file that contains email addresses to be loaded into the global address book. This column must be a STRING, CSTRING, or PSTRING column type.

Name Field Specifies a column from a data file that contains the name corresponding to the email address from the **Address Field**. This column must be a STRING, CSTRING, or PSTRING column type.

Default Class Select a class from the drop down list box. The class defines what group the email address belongs to. Valid classes are *Send To* (primary recipient), *CC* (secondary recipient), and *BCC* (hidden recipient).

Transport Type Specifies the default transport protocol to use when sending email. The default value is SMTP (Simple Mail Transport Protocol). This may be a string, variable, or runtime expression using EVALUATE. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

FileTriggers

The FileTriggers extension template works together with the GlobalDocumentHandler and DbAuditing templates to trigger an email when an update occurs on a data file. The GlobalDocumentHandler and DbAuditing templates must be present in the application in order to use the FileTriggers extension. Each file in the dictionary can have its own defined triggers and settings.

Trigger Settings A list box is provided which lists all tables defined in the dictionary associated with the application. If a table has a trigger defined, it is noted in the list box. To modify the trigger settings, highlight the file and click on the **Properties** button.

Triggers Check the box to activate a trigger on a file update. Triggers respond on Insert, Change and Delete file updates.

Change Trigger
The Change Trigger tab is displayed by checking the **Changes?** box above. This tab configures the email message to be sent when a record changes in the specified data file.

Send Trigger Specifies the conditions that trigger an email. Choose one of the following from the drop down list.

Per Column Change
Triggers email when one or more of the Fields to Watch change. A separate email is sent for each of the watch fields. The email is sent when the changes are confirmed.

Per Row Change
Triggers email when any field in the table changes. One email is sent regardless of how many fields change.

Subject Specifies a string, variable, or runtime expression using EVALUATE which contains the subject to use in the triggered email. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Content Type Specifies the type of email to be sent. Choose either *Html* or *Plain Text* from the drop down list.

Message Body Source
Specifies the body content of the triggered email. The body can be information inserted from the contents of a file or can be the contents of a column in a table. Choose *Field* or *File* from the drop down list.

File
Choose a file to use in the message body by pressing the ellipsis (…) button. This option is available when the **Message Body Source** specifies *File*.

Use Skeleton to merge document
Check this box to to merge the chosen file with a skeleton html file (skeletons are primarily used for Web Builder applications) to provide personalized information in the body text of the message.

Field
Choose a field to use in the message body by pressing the ellipsis (…) button. This option is available when the **Message Body Source** specifies *Field*.

Recipients
Specifies the destination of the triggered email. One or more recipients may be specified.

Address
Specifies a string, variable, or runtime expression using EVALUATE which contains an email address to which the email is sent. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Name
Specifies a string, variable, or runtime expression using EVALUATE which contains a users name to which the email is sent. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Recipient Class
Choose from the drop down list which type of recipient you are specifying. Choose *Send To* for a primary recipient. Choose *CC* for a secondary recipient. Choose *BCC* for a hidden recipient.

Fields to Watch
Specifies the columns that trigger an email. This option is only available if *Per Column Change* is selected as the send trigger. Press the **Insert**, **Properties**, or **Delete** buttons to update this field list.

FieldName
Choose a column that triggers an email by pressing the ellipsis (…) button

Subject Specifies a string, variable, or runtime expression using EVALUATE which contains the subject to use in the triggered email. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Content Type Specifies the type of email to be sent. Choose either *Html* or *Plain Text* from the drop down list.

Message Body Source
Specifies the body content of the triggered email. The body can be information inserted from the contents of a file or can be the contents of a column in a table. Choose *Field* or *File* from the drop down list.

File Choose a file to use in the message body by pressing the ellipsis (…) button. This option is available when the **Message Body Source** specifies *File*.

Use Skeleton to merge document
Check this box to merge the chosen file with a skeleton html file (skeletons are primarily used for Web Builder applications) to provide personalized information in the body text of the message.

Field Choose a field to use in the message body by pressing the ellipsis (…) button. This option is available when the **Message Body Source** specifies *Field*.

Insert Trigger
The Insert Trigger tab is displayed by checking the **Inserts?** box above. This tab configures the email message to be sent when a record is inserted in the specified data file.

Subject Specifies a string, variable, or runtime expression using EVALUATE which contains the subject to use in the triggered email. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Content Type Specifies the type of email to be sent. Choose either *Html* or *Plain Text* from the drop down list.

Message Body Source

Specifies the body content of the triggered email. The body can be information inserted from the contents of a file or can be the contents of a column in a table. Choose *Field* or *File* from the drop down list.

File

Choose a file to use in the message body by pressing the ellipsis (…) button. This option is available when the **Message Body Source** specifies *File*.

Use Skeleton to merge document

Check this box to merge the chosen file with a skeleton html file (skeletons are primarily used for Web Builder applications) to provide personalized information in the body text of the message.

Field

Choose a field to use in the message body by pressing the ellipsis (…) button. This option is available when the **Message Body Source** specifies *Field*.

Recipients

Specifies the destination of the triggered email. One or more recipients may be specified.

Address

Specifies a string, variable, or runtime expression using EVALUATE which contains an email address to which the email is sent. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Name

Specifies a string, variable, or runtime expression using EVALUATE which contains a users name to which the email is sent. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Recipient Class

Choose from the drop down list which type of recipient you are specifying. Choose *Send To* for a primary recipient. Choose *CC* for a secondary recipient. Choose *BCC* for a hidden recipient.

Delete Trigger

The Delete Trigger tab is displayed by checking the **Deletes?** box above. This tab configures the email message to be sent when a record is deleted in the specified data file.

Subject
Specifies a string, variable, or runtime expression using EVALUATE which contains the subject to use in the triggered email. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Content Type
Specifies the type of email to be sent. Choose either *Html* or *Plain Text* from the drop down list.

Message Body Source
Specifies the body content of the triggered email. The body can be information inserted from the contents of a file or can be the contents of a column in a table. Choose *Field* or *File* from the drop down list.

File
Choose a file to use in the message body by pressing the ellipsis (…) button. This option is available when the **Message Body Source** specifies *File*.

Use Skeleton to merge document
Check this box to merge the chosen file with a skeleton html file (skeletons are primarily used for Web Builder applications) to provide personalized information in the body text of the message.

Field
Choose a field to use in the message body by pressing the ellipsis (…) button. This option is available when the **Message Body Source** specifies *Field*.

Recipients
Specifies the destination of the triggered email. One or more recipients may be specified.

Address
Specifies a string, variable, or runtime expression using EVALUATE which contains an email address to which the email is sent. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).

Name	Specifies a string, variable, or runtime expression using EVALUATE which contains a users name to which the email is sent. To specify a variable here, precede the entry with an exclamation point (!). To specify a runtime expression, precede the entry with an equal sign (=).
Recipient Class	Choose from the drop down list which type of recipient you are specifying. Choose *Send To* for a primary recipient. Choose *CC* for a secondary recipient. Choose *BCC* for a hidden recipient.

Threading Template Support

Clarion 6 offers the following templates that control the default Thread Model:

Global Cooperative Threading Extension

Options
The options for this extension are made up of a single check box.

Enable Global Cooperative Threading

Check this box to switch your application to use a cooperative threading model in place of the preemptive model.

More information on these models (including advantages and disadvantages) can be found in the *Preemptive and Cooperative Thread Models*

This template automatically includes a *Preemptive Procedure Extension* in all procedures defined in your application. This allows you exact and convenient control over each procedure's thread models.

For example, if the majority of procedures will need to use the cooperative thread model, check the global option, and use the Preemptive Procedure Extension to enable the few exceptions. On the contrary, you can turn off the Global Cooperative Threading model, which makes all procedures preemptive by default, and enable Cooperative Threading by turning off the individual procedure's Preemptive Procedure Extension.

Preemptive Procedure Extension

The Preemptive Procedure Extension is included in each of your application's procedures if you have included the *Global Cooperative Threading* Extension template in your application.

The sole purpose of this template is to provide an override to the Global settings. If the Cooperative threading model is active, check this box to allow this procedure to use the Preemptive model instead. On the contrary, if the cooperative threading model is turned off on the global level, turn off the check box here to allow this procedure to use the cooperative threading model.

More information on these models can be found in the *Preemptive and Cooperative Thread Models* documentation.

Viewer (ASCII) Based Templates

ASCII Print Button control template

This Control template adds a button named Print, and the underlying code necessary for printing an ASCII (text) file. Use this control template together with the ASCII View control template.

Edit the **Actions** only if you wish to add another, separate action to take place *after* printing. All the code necessary for managing the print job itself is handled automatically.

The **Actions** tab contains the following:

When Pressed	The standard set of prompts for buttons. Normally, when using a Control template, these prompts are not used.

ASCII Search Button control template

This Control template adds two buttons named Find and Find Next, and the underlying code necessary for a modal search dialog, allowing the end user to find text in an ASCII (text) file. Use this control template together with the ASCII View control template .

Edit the **Actions** only if you wish to add another, separate action to take place *after* the search. All the code necessary for managing the search itself is handled automatically.

The **Actions** tab contains the following:

When Pressed	The standard set of prompts for buttons. Normally, when using a Control template, these prompts are not used.

ASCII View control template

The AsciiViewControl template adds a LIST control in which you can display read-only, the contents of a file--including variable length files. It is typically used to display an ASCII text file. The AsciiViewControl template optionally provides search and print capability for the displayed file.

The template lets you select the file to view at design time, or leaves the selection to the end user at runtime if you prefer. Finally, the template optionally allows the LIST control to alternate its display between the selected file and some other data that you specify.

The AsciiViewControl template provides embed points for its LIST control. It also provides the following prompts on the **List Properties** dialog **Actions** tab, the **Procedure Properties** dialog, or the **Extension and Control Templates** dialog:

General Options

Initialize Viewer

Determines when the procedure initializes the Viewer object. Initialization includes selecting the file to view, opening it, and reading it.

On Open Window

Initializes the Viewer when the window opens so that the Viewer's LIST is full upon initial display.

On Field Selection

Delays initializing the Viewer until the end user selects the Viewer's LIST control.

Manually

Does not initialize the Viewer. You must embed a call to the Viewer#.Initialize ROUTINE to initialize the Viewer.

File to Browse

Specifies the path and name of the file to view, or a variable containing the path and name of the file to view. The variable must be preceded by an exclamation point (!).

If no path is specified, the procedure looks for the file in the current directory.

If omitted (left blank), the Viewer object prompts the end user to select a file.

Reassign FROM attribute after Kill

Check this box to reset the Viewer LIST's FROM attribute after the Viewer shuts down. See *FROM* in the *Language Reference*. This lets you use a single LIST control to display both the **File to Browse** and other items as well.

Value or queue to assign

Type the label of the QUEUE (or the string constant) to assign to the Viewer LIST's FROM attribute.

Allow popup menu searching

Check this box to provide a (RIGHT-CLICK) popup menu choice to search the file.

Allow popup menu printing

Check this box to provide a (RIGHT-CLICK) popup menu choice to print some or all of the records in the file.

Classes Tab

Use the Classes tab to override the global settings for the Class. See Classes Tab.

See also:

ASCII Print Button

ASCII Search Button

ASCII View in List box extension

ASCII View in List box

The AsciiViewInListBox template allows a LIST control to alternate its display between a selected file and some other data that you specify.

The AsciiViewInListBox template provides the same functionality and the same prompts as the AsciiViewControl template. See *AsciiViewControl* for more information. The AsciiViewInListBox template provides one additional prompt. Because it is an Extension template and does not place its own contol, the AsciiViewInListBox template prompts you for the LIST control to use to display text:

General Tab

List box field to use

Select the LIST control that alternates its display.

Initialize Viewer

Determines when the procedure initializes the Viewer object. Initialization includes selecting the file to view, opening it, and reading it.

On Open Window

Initializes the Viewer when the window opens so that the Viewer's LIST is full upon initial display.

On Field Selection

Delays initializing the Viewer until the end user selects the Viewer's LIST control.

Manually

Does not initialize the Viewer. You must embed a call to the Viewer#. Initialize ROUTINE to initialize the Viewer.

File to Browse

Specifies the path and name of the file to view, or a variable containing the path and name of the file to view. The variable must be preceded by an exclamation point (!).

If no path is specified, the procedure looks for the file in the current directory.

If omitted (left blank), the Viewer object prompts the end user to select a file.

Allow popup menu searching

Check this box to provide a (RIGHT-CLICK) popup menu choice to search the file.

Allow popup menu printing

Check this box to provide a (RIGHT-CLICK) popup menu choice to print some or all of the records in the file.

Classes Tab

Use the Classes tab to override the global settings for the Class. See Classes Tab.

Wizard and Utility Templates

This section covers the powerful Wizard and Utility templates that are shipped with Clarion. For more information, see the *Template Language Reference* PDF.

Application Wizard utility template

This wizard creates a complete application from an existing dictionary. It creates a Frame containing a menu with options calling all procedures it creates. It also creates Browse and Report procedures for each specified file, with associated Form (Update) procedures.

Answer the questions in each dialog, then press the Next button.

After the introduction screen, you are presented with the following options:

Theme Selection:

Theme
Select from the drop list of themes. Themes are groups of settings that control colors, fonts, icons, backgrounds, positions and much more - for Frame, Browse, Form and Report procedures. You will have the opportunity to create a new theme as you progress through the wizard. Select a starting or default theme here.

Report Layout
Select a default report layout from the drop list provided. This layout will be the basis for all of the reports that will be generated by the wizard.

If you are using the **Quick Start Wizard**, this is the only dialog that you will be presented with. Press the Finish button to begin the creation of your application.

Other prompts that follow on subsequent windows:

Generate Procedures for all files in my dictionary
Check the box for all tables, or clear the box to select specific tables.

For SQL based file systems, the Application Wizard also generates code to capture user login information upon initial program load, and then reuse the login information for each file accessed.

Which control model should the Application use?

There are three models the wizard can use to create applications: Button, Toolbar, or Both.

Button
The wizard builds the application with traditional Insert, Change, Delete, OK, and Cancel command buttons that appear on each dialog.

Toolbar
The wizard builds the application with global toolbar command buttons that appear on the application frame. The toolbar buttons control each dialog. See Control Templates-FrameBrowseControl for more information.

Both
The wizard builds the application with both the traditional dialog command buttons and the global toolbar command buttons.

Customization

Wizards have different "look and feel" settings and actions called *themes*, which can be modified and saved for use in other applications. Themes are set and controlled by a variety of customization options.

Press the **Next** button to accept the selected theme's settings, or press one of the customization buttons to modify them at this time.

Create an Internet Enabled Application
Check this box to apply the Web Extension templates to your application. This allows you application to be deployed as both a Windows and Internet application.

Overwrite existing procedures
Check this box to overwrite existing procedures with the same names. Clear the box to preserve existing procedures.

Generate Reports for each file
Check this box to automatically generate report procedures. Clear the box to omit report procedures.

Select Sort Order
Select from the drop list the sort and report generation method from the following choices:

Single Key
Select this option to force the wizard to generate a separate report for each key defined in your file (or files).

Runtime Key Selection
Select this option to force the wizard to generate a single report that pops up a sort order dialog prior to printing at runtime.

Record Order
Select this option to force the wizard to generate a single report sorted by record order for your selected file (or files).

On the last dialog, the **Finish** button is enabled. If you are satisfied with your answers, press the **Finish** button. You also have the option here to **Save Changes**, where any changes to customization options are saved to the theme that you selected at the start of the wizard. If you wish, you can opt to **Save on a new theme,** and enter the new name of the Theme and Theme file**.**

You can press the **Back** button to change a prior selection or press the **Cancel** button to abandon the application.

The Application Wizard creates the .APP file based on the dictionary and the answers you provided, and then displays the Application Tree dialog for your new application.
Fine Tuning the Wizard

You can control how the wizard builds your application by specifying options for Tables, Columns, Keys, and Relationships in the Data Dictionary (see Dictionary Options Dialog).

Clarion Quick Start and Dictionary Quick Load

Using the Quick Start Wizard, you can create a data dictionary and a working application with no coding required.

Simply define a data table and the Quick Start Wizard creates a complete Windows application--in about five minutes if you're a fast typist! Your application has a form procedure for updating the table, and as many view windows and reports as the data table has keys.

Just define the columns for a single table. For each column, you provide a name, display format picture, and key information. This creates a data dictionary. The Quick Start Wizard creates the application based on this dictionary. Once you've specified all options, the **OK** button generates the .APP file, and loads the procedures into the **Application Tree** dialog.

The Quick Load Wizard is similar to the Quick Start Wizard; the only difference is that its function is exclusively to create a data table definition as an addition to an existing data dictionary. After creating the table definition, you can use one of the Procedure Wizards to create procedures using the table.

You can call the Quick Load Wizard by pressing the **Add Table** button in the **Dictionary** dialog. Once you've specified all the options, the **OK** button adds a new table definition to the **Dictionary** dialog, complete with **Column/Key Definitions**.

Application Name

Type a legal DOS file name for the .APP file. The Quick Start Wizard will use the same file name (with the .DCT extension) for the data dictionary file.

Optionally press the ellipsis button (...) to change the directory, and type a file name in the Open File dialog box. The working directory, in which all source code files will be generated, depends on where the .APP file resides.

Because the Quick Load Wizard does not create the .APP file, this control is not present in the Quick Load Wizard.

Data Table Name

Type a legal DOS file name (no extension necessary) for the data table.

Prefix

This box automatically fills in with the first three letters of the name of the data file when you TAB away from the **Data Table Name** box. Optionally specify up to three letters of your choice in this field.

The prefix allows your application to distinguish between similar variable names occurring in different table 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 distinguish the two columns as ORD:INVOICE and SAL:INVOICE.

Table Driver

Specify the data table type. When using the Application Generator, Clarion automatically links in the correct database table driver library. See the *Database Drivers* topic for a discussion of the relative advantages of each driver.

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.

Column Name

To name each column, type a valid Clarion label in the **Name** field. Valid column names may vary slightly according to the database driver.

The Quick Start Wizard allows you to name each column, one by one, by pressing the DOWN ARROW to add a new item to the list. Before naming the next column, specify the **Picture** and **Key** options for the current key.

Picture

Specify a default picture token by typing it in the **Picture** field. The picture token, together with the selected **File Driver**, determine the data type which the Quick Start Wizard uses for the column. When the Application Generator creates window and report controls for the column, this also serves as the default picture for the control.

The Quick Start Wizard allows you to name each column, one by one, by pressing the DOWN ARROW to add a new item to the list. Before naming the next column, specify the **Key** option for the current key.

Key

This specifies whether to create a key using this column as a component, and if so, the type of key. By specifying **Unique***,* your application will ensure that each record has a distinct key value. **Duplicate** specifies a key that allows more than one record with the same value in the key component. **Autonumber** creates a key with an auto-incrementing value. The Application Generator creates code to manage record sequence numbers.

The Quick Start Wizard creates a multi-keyed browse procedure and reports for every key you specify.

The Quick Start Wizard allows you to name each column, one by one, by pressing the down arrow to add a new item to the list. Press the DOWN ARROW, or TAB, to define the next column.

Insert This button allows you to insert a new, blank column, above the currently selected column.

Delete This button allows you to delete the currently selected column.

Move Up This button allows you to move the currently selected column up one position in the columns list.

Move Down This button allows you to move the currently selected column down one position in the columns list

Dictionary Print Wizard utility template

This wizard prints a description of an existing dictionary file at varying levels of detail for files, fields, keys, and relationships. You may print to the printer or to a file.

To use the Dictionary Print Wizard:

1. Open an application that uses the dictionary.
2. Choose **Application ▸ Template Utility** from the menu.

   The **Select Utility** dialog appears.
3. Highlight **DictionaryPrint** and press the **Select** button.

   The **Dictionary Print Wizard** dialogs appear.
4. Answer the question(s) in each dialog, then press the **Next** button.

   After the first dialog, the **Finish** button is enabled. Press the **Finish** button now to print all the information available for all the files, fields, keys, and relationships.

   Or, step through the wizard's dialogs, to select specific files, plus the level of detail to print (All, Some, or None) for the various dictionary components.

Label Wizard utility template

This wizard creates a report procedure from an existing dictionary file definition that includes a defined report label layout.

After the introduction screen, you are presented with the following options:

Theme Selection

Theme	Select from the drop list of themes. Themes are groups of settings that control colors, fonts, icons, backgrounds, positions and much more - for Frame, Browse, Form and Report procedures. You will have the opportunity to create a new theme as you progress through the wizard. Select a starting or default theme here.
Label Group	Select a label group from the drop list. A label group contains the most popular classes of labels (Avery, Card Products, InkJet, etc.). If your type of label is not listed here, select **Others**.
Label Type	Select a label type from the drop list. A label type normally corresponds to its product code.
Save Settings	After you have selected a theme, you have the option to save these settings for any future applications that you create.

What name should be used as the label of the report procedure?

Type the procedure name.

Which file do you want to report?

Press the ellipsis (...) button to select a file from the dictionary.

A report can use a single record key, or can run in record order. Enter a key below, or leave the field blank to run in record order.

Press the ellipsis (...) button to select a sort key. Leave the field blank to specify no sort key.

Add label string to each field?

Check this box if you wish to include a report header string that identifies the label fields used in the report.

Select the fields that you want to use

Build your report in this list box by adding and deleting fields from the selected file. You can also modify the properties of the fields regarding column labels, picture tokens, and justification. Use the arrow buttons to specify the order that each field will appear on the report.

Customization

Wizards have different "look and feel" settings and actions called *themes*, which can be modified and saved for use in other applications. Themes are set and controlled by a variety of customization options.

Press the **Next** button to accept the selected theme's settings, or press the customization button to modify it at this time.

Overwrite existing procedures

Check this box to overwrite existing procedures with the same names. Clear the box to preserve existing procedures.

On the last dialog, the **Finish** button is enabled. If you are satisfied with your answers, press the **Finish** button. You also have the option here to **Save Changes**, where any changes to customization options are saved to the theme that you selected at the start of the wizard. If you wish, you can opt to **Save to a new theme.**

List Help IDs (ListHLPIDs) Utility Template

The ListHLPIDs utility template searches through an active application and extracts all help IDs specified by any existing HLP attribute on a WINDOW or control.

If a WINDOW HLP ID is found, the procedure name is extracted and exported with the Help ID to a designated text file. The Field Equate Label is also extracted for controls that have a Help ID assigned.

The following prompts are provided:

Output File

Press the ellipsis button to select a file to generate output to, or enter a new name in the entry field provided.

Skip Items with no HLP attribute

Check this box to only generate procedures and controls that have an HLP attribute attached to them. This could be very convienent for large applications.

Example Text Output:

```
********* List of All HLP IDs for appdemoa *********

Application Help File: myhelp.hlp

  - HTML Help is Enabled for this Application
    Default Help File Name: cwhh.chm
    Append .HTM to Help IDs is ON

  WALink                !procedure name
    ?sSearch:Prompt     !control feq
    ?sSearch
    ?ButtonLookup
    ?String1

  wKeyword  -  'testid.htm'   !procedure with HLP attribute
    ?sSearch:Prompt
    ?sSearch  -  'test2.htm'  !control with HLP attribute
    ?ButtonLookup
    ?String1
```

Note:

The leading tilde (~) is always stripped out of the generated output. The .HTM extension is visible when set by the Global HTML Help Template.

Process Wizard utility template

This wizard creates a process procedure from an existing dictionary file definition.

After the introduction screen, you are presented with the following options:

Theme Selection

Theme	Select from the drop list of themes. Themes are groups of settings that control colors, fonts, icons, backgrounds, positions and much more - for Frame, Browse, Form and Report procedures. You will have the opportunity to create a new theme as you progress through the wizard. Select a starting or default theme here.
Save Settings	After you have selected a theme, you have the option to save these settings for any future applications that you create.

What name should be used as the label of the Process procedure?

Type the procedure name.

Which file do you want to process?

Press the ellipsis (...) button to select a file from the dictionary.

A process can use a single record key, or can run in record order. Enter a key below, or leave the field blank to run in record order.

Press the ellipsis (...) button to select a sort key. Leave the field blank to specify no sort key.

Customization

Wizards have different "look and feel" settings and actions called *themes*, which can be modified and saved for use in other applications. Themes are set and controlled by a variety of customization options.

Press the **Next** button to accept the selected theme's settings, or press one of the customization buttons to modify them at this time.

Overwrite existing procedures

Check this box to overwrite existing procedures with the same names. Clear the box to preserve existing procedures.

On the last dialog, the **Finish** button is enabled. If you are satisfied with your answers, press the **Finish** button. You also have the option here to **Save Changes**, where any changes to customization options are saved to the theme that you selected at the start of the wizard. If you wish, you can opt to **Save to a new theme.**

ViewABCClasses Utility Template

This utility 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.

Select the appropriate tab for the type of display.

Methods and properties are color-coded to let you know if they are PRIVATE, PROTECTED, or VIRTUAL.

To locate a specific Methods and property, press the **Find** button.

Optionally, you can filter the list by clearing the **Show Private** and/or the **Show Protected** checkboxes.

This viewer only lists ABC Compliant Classes. See *ABC Compliant Classes* for more information.

Find What	Type in the search string.
Direction.	Set the direction option you want to control the search.
Find	Press this button to start the search.

Window Wizard utility template

This wizard creates a window using template themes and other basic settings.

After the introduction screen, you are presented with the following options:

Theme Selection

Theme Select from the drop list of themes. Themes are groups of settings that control colors, fonts, icons, backgrounds, positions and much more - for Frame, Browse, Form and Report procedures. You will have the opportunity to create a new theme as you progress through the wizard. Select a starting or default theme here.

Save Settings After you have selected a theme, you have the option to save these settings for any future applications that you create.

What name should be used as the label of the procedure?

Type the window procedure name.

Customization

Wizards have different "look and feel" settings and actions called *themes*, which can be modified and saved for use in other applications. Themes are set and controlled by a variety of customization options.

Press the **Next** button to accept the selected theme's settings, or press the Window Customization button to modify it at this time.

Overwrite existing procedures

Check this box to overwrite existing procedures with the same names. Clear the box to preserve existing procedures.

On the last dialog, the **Finish** button is enabled. If you are satisfied with your answers, press the **Finish** button. You also have the option here to **Save Changes**, where any changes to customization options are saved to the theme that you selected at the start of the wizard. If you wish, you can opt to **Save to a new theme.**

Wizard Themes

The Application Generator is packed with special template-based *wizards*, which are designed to help you generate rich, full-featured applications or individual procedures, based on a series of simple information prompts and options.

These wizards also have different "look and feel" settings and actions called *themes*, which can be modified and saved for use in other applications. Themes are set and controlled by a variety of customization options.

Frame Customization

Browse Customization

Form Customization

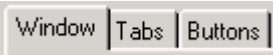
Report Customization

Window Customization

Process Customization

Theme Maintenance

Browse Customization



The general appearance of the Browse Wizard is controlled by the settings on this window. These settings can be saved in a "theme" for use in future applications.

Procedure Name

Enter a name that the Browse Wizard will use to generate procedure names in your application. The template macro symbol, %FileName, is required, and extracts the Dictionary file name for each file selected by the wizard. You can modify this line with other template macros and text if you wish. No spaces are allowed in procedure names.

Browse Message

Enter default text that the Browse Wizard will generate in the MSG attribute of each list box. The user will see this text in the status bar of the runtime application when the list box is selected. The %FileName macro is not required here, but recommended.

Window

Caption
Text that is entered on this line will be used as the description of the window used by the Browse Wizard.

Secondary Caption
Text that is entered on this line will be used as the message that appears in the status bar of a window only when the browse box is populated as a secondary file on a Form.

Images:
Background
Enter a default image here to use as a graphic or watermark for your Browse window.

Mode
If you have designated a background image to use for your Browse window, this option becomes available to control if the image is tiled, stretched, or centered.

Icon
Enter a default icon to use for your Browse procedure’s window. This will allow your Browse window to be minimized if needed.

Font
Press the Font button to select a default font to use for the Browse procedure. Sample text shown below the button is provided to allow you to review your selection.

Options
Select from the drop list to designate the initial position of your Browse window. You can center the window, or use the default position that is set by the template wizard.

In addition, click on the **System Menu** check box to add the Windows System Menu to your Browse procedure. If you will be using any entry fields on the Browse window, you can also click on the **Entry Patterns** check box to allow special formatting picture information (Example: phone numbers or date pictures)

Tabs

Tab Text
Enter text that will be used for each tab control that is generated by the Browse wizard. The default setting is ***&%#) %Key.*** The ampersand (&) identifies the next character as the hot key of the tab control. The **%#** macro identifies the instance or order of the key used in the tab control. The %Key is the description of the key name (or label of the key if the description in the Dictionary Editor is blank).
Example: *2) By Account*

Sort Order Selection Style
Press the **Select Style** button to access the Select Style dialog window.

Buttons

The Buttons tab is the central control for all buttons used on the window. There is a default **Width** and **Height** setting (in dialog units) used for all buttons. However, each button can be overridden individually through the Button Customization dialog.

The default buttons of the Browse wizard are:
Insert
Change
Delete
View
Select
Close
Parent Select
Help

The generation of each button is controlled by other settings within the template wizard. However, you can also override the generation of the **View** and **Help** buttons by unchecking the appropriate boxes.

Browse Sort Order Customization

This window is used to set the sort order style of a default Browse procedure. Sort order style is defined as the way you navigate from one sort value to the next based on the number of keys that you designate for the Browse. You can select from the following options:

Which selection style?

Choose from *Tabs*, *Drop List*, or *Popup menu*.

The *Tabs* selection uses tab controls to signal the list box that a new sort order is to be applied. The *Drop List* option populates a drop list control that contains the sort order selections of the selected Browse Box control. The *Popup menu* option stores the sort order selections in a popup menu, which is accessed when right-clicking on the Browse Box (List Box) control, or pressing the Sort Order button that is created. In addition, *Popup menu* enables the following additional options for the Sort Order button:

Text
Enter the default text to display on the Sort Order button.

Icon
Enter an icon file or equate that you wish to use for the Sort Order button.

Cursor
Enter a default cursor file or equate that you wish to use for the Sort Order button. This cursor will be displayed when the mouse moves over the button area.

HotKey
Enter a default hot key to use to automatically call the Sort Order button.

Message
Enter a default message to use for the Sort Order button. This text will appear by default in the window's status bar.

Tip
Enter a tool tip message for the Sort Order button. This text will appear when the mouse is moved over the button.

Options
Activate the **Flat** check box to give the Sort Order button a flat appearance.

Activate the **Skip** check box to disable tabbing to the Sort Order button.

Select a **Justification** mode to designate where the icon (if any) will appear in relation to the button text.

Button Customization

This window contains the generic settings used for selected buttons generated by the wizards. The title of the window reflects which button's properties is currently being modified (i.e., Insert Button, Close Button, etc.).

Common properties include:

Procedure Name (Only valid for the Parent Select button)

Enter a procedure name that the button will call to select a parent record. This button is normally displayed in Browse procedures that use a Child file as the primary file. The template macro symbol, %FileName, is required, and extracts the Dictionary file name for the parent file that is related to the child procedure. You can modify this line with other template macros and text if you wish. No spaces are allowed in procedure names.

Child Browse Proc Name (Only valid for the Child button)

Enter a procedure name that the button will call to browse a child file related to the active primary file. This button is normally displayed in Form procedures to browse a Child file's records to view or modify. The template macro symbol, %FileName, is required, and extracts the Dictionary file name for the child file that is related to the Form's parent. You can modify this line with other template macros and text if you wish. No spaces are allowed in procedure names.

Window Name (Valid for the Parent Select and Child Buttons)
Text that is entered on this line will be used as the description of the window used by the procedure that is called by the button.

Position
Choose from the drop list an approximate location to place this button, or select **Other** from the drop list to set custom position parameters.

X
Enter an integer constant that specifies the horizontal position of the top left corner of the control.

Y
Enter an integer constant that specifies the vertical position of the top left corner of the control.

Width
Enter an integer constant that specifies the width of the specified control

Height
Enter an integer constant that specifies the height of the specified control.

Text
Enter the text to display on the button

Icon
Enter an icon file name, or use the ellipsis button to select an icon file that you wish to use for this button.

Cursor
Enter a cursor file name, or use the ellipsis button to select a cursor file that you wish to use for this button. The cursor will be displayed when the mouse moves over the button area.

HotKey
Enter a Clarion keycode or equate, or press the ellipsis button to select a key to use as the hot key for this button. Pressing the hot key at program runtime should activate the button.

Message
Enter a message to display in the status bar by default when the mouse moves over the button area.

Tip
Enter a tool tip message to display when the mouse moves over the button area.

Options
Activate the **Flat** check box to give the selected button a flat appearance.

Activate the **Skip** check box to disable tabbing to the selected button.

Select a **Justification** mode to designate where the icon (if any) will appear in relation to the button text.

Window Control Customization

Throughout parts of the Clarion template wizards, there are default window controls that are automatically populated as part of a generated procedure. Each control has settings for the

X and Y, width and height parameters as follows:

X
Enter an integer constant that specifies the horizontal position of the top left corner of the control.

Y
Enter an integer constant that specifies the vertical position of the top left corner of the control.

Width
Enter an integer constant that specifies the width of the specified control

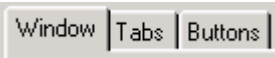
Height
Enter an integer constant that specifies the height of the specified control.

Justification
Select a **Justification** mode to designate how the text will appear in relation to the control's position.

Options

For Progress controls, you also have the option to designate a **vertical** (bottom to top) orientation, and a **smooth** bar display.

Form Customization



The settings on this window control the appearance of the Form procedure. These settings can be saved for use in other future applications.

Procedure Name

Enter a name that the Form Wizard will use to generate procedure names in your application. The template macro symbol, %FileName, is required, and extracts the Dictionary file name for each file selected by the wizard. You can modify this line with other template macros and text if you wish. No spaces are allowed in procedure names.

Window

Caption
Text that is entered on this line will be used as the description of the window used by the Form Wizard.

Background
Enter a default image here to use as a graphic or watermark for your Form window.

Mode
If you have designated a background image to use for your Form window, this option becomes available to control if the image is tiled, stretched, or centered.

Icon
Enter a default icon, or press the ellipsis button to select an icon file for use in your Form procedure's window. This will allow your Form window to be minimized if needed.

Font
Press the Font button to select a default font to use for the Form procedure. Sample text shown below the button is provided to allow you to review your selection.

Options
Select from the drop list to designate the initial **position** of your Form window. You can center the window, or use the default position that is set by the template wizard.

In addition, click on the **System Menu** check box to add the Windows System Menu to your Form procedure. If you will be using any entry fields on the Form window, you can also click on the **Entry Patterns** check box to allow special formatting picture information (Example: phone numbers or date pictures)

The **Entry Mode** drop list allows you to set the typing mode for the Form. 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.

Tabs

Form Tabs

Enter text that will be used for each tab control that will be generated by the Form wizard. The default setting is ***&%#) %Text.*** The ampersand (&) identifies the next character as the hot key of the tab control. The **%#** macro identifies the instance number of the tab generated, determined by the number of fields per tab control and the total number of fields populated. The %Text is a declared template symbol that defaults to "General". You can remove this symbol and substitute it with any text that you wish.
Example: *2) General*

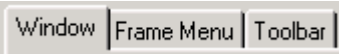
Buttons

The Buttons tab is the central control for all buttons used on the window. There is a default **Width** and **Height** setting (in dialog units) used for all buttons. However, each button can be overridden individually through the Button Customization dialog.

The default buttons of the Form wizard are:

OK
Cancel
Child
Help

Frame Customization



The settings on this window control the appearance of the Frame procedure. These settings can be saved for use in other future applications.

Window

Background
Enter a default image here to use as a graphic or watermark for your Frame procedure's client area (the area below your menu and toolbar by default).

Mode
If you have designated a background image to use for your Frame's client area, this option becomes available to control if the image is tiled, stretched, or centered.

Icon
Enter a default icon to use for your Frame procedure's window. This will allow your default applications to be minimized if needed.

Font
Press the Font button to select a default font to use for the Frame procedure. Sample text shown below the button is provided to allow you to review your selection.

Position
Select from the drop list to designate the initial position of your Frame window. You can center the window, or use the default position that is set by the template wizard.

In addition, click on the **System Menu** check box to add the Windows System Menu to your Frame procedure. If you will be using any entry fields on the Frame window's tool bar, you can also click on the **Entry Patterns** check box to allow special formatting picture information (Example: phone numbers or date pictures)

FrameMenu

File

Enable File Menu
Check this box if you would like the wizard to create a File menu in your Frame procedure. In addition, you can customize the following default menu items and their associated messages (which are displayed in the Frame's status bar by default):

File Menu Text
Print Item Text
Print Item Msg
Exit Item Text
Exit Item Msg

Edit

Enable Edit Menu
Check this box if you would like the wizard to create an Edit menu in your Frame procedure. In addition, you can customize the following default menu items and their associated messages (which are displayed in the Frame's status bar by default):

Edit Menu Text
Cut Item Text
Cut Item Msg
Copy Item Text
Copy Item Msg
Paste Item Text
Paste Item Msg

Window

Enable Window Menu
Check this box if you would like the wizard to create a standard Window menu in your Frame procedure. In addition, you can customize the following default menu items and their associated messages (which are displayed in the Frame's status bar by default):

Window Menu Text
Tile Item Text
Tile Item Msg
Cascade Item Text
Cascade Item Msg
Arrange Icons Item Text
Arrange Icons Item Msg

Help

Enable Help Menu
Check this box if you would like the wizard to create a standard Window menu in your Frame procedure. In addition, you can customize the following default menu items and their associated messages (which are displayed in the Frame's status bar by default):
Help Menu Text
Contents Item Text
Contents Item Msg
Search Item Text
Search Item Msg
How Item Text
How Item Msg

Toolbar

General

Background
Enter a default image here to use as a graphic or watermark for your Frame's toolbar.

Mode
If you have designated a background image to use for your Frame's toolbar, this option becomes available to control if the image is tiled, stretched, or centered.

Height
Enter the height in dialog units for the Frame procedure toolbar.

Tip

This tab identifies the toolbar buttons that are available from the wizard. You can modify the tool tip message for each button here.

Icon Properties

Width
The Frame procedure wizard has the option to produce toolbar navigation buttons. Enter here the default width of each button in dialog units.

Height
The Frame procedure wizard has the option to produce toolbar navigation buttons. Enter here the default height In dialog units of each button. Care must be taken not to exceed the default height of the overall toolbar properties.

Flat
Activate this checkbox to produce a flat button appearance for all toolbar buttons.

Icon

This tab identifies the toolbar buttons' default icon files used by the wizard. You can modify the icon files for each button here.

Process Customization

The settings here control the appearance of the Process template procedure generated by the Process Wizard. These settings can also be saved for use in other future applications.

Procedure Name
Enter a name that the Process Wizard will use to generate as a procedure name in your application. The template macro symbol, %FileName, is required, and extracts the Dictionary file name for each file used by the wizard. You can modify this line with other template macros and text if you wish. No spaces are allowed in procedure names. This name is used for files that have no keys defined

Proc. "With Key" Name
This prompt is similar to Procedure Name, but is used for designated files that have a key (or keys) defined in the Dictionary Editor.

Progress Window Name
Text that is entered on this line will be used as the description of the window used by the Process Wizard.

Background
Enter a default image here to use as a graphic or watermark for your Process window.

Mode
If you have designated a background image to use for your Process window, this option becomes available to control if the image is *Tiled*, *Stretched*, or *Centered*.

Icon
Enter a default icon, or press the ellipsis button to select an icon file for use in your Process procedure's window. This will allow your Form window to be minimized if needed.

Font
Press the Font button to select a default font to use for the Process window and controls. Sample text shown below the button is provided to allow you to review your selection.

Options
Select from the drop list to designate the initial **Width, Height ,** and **Position** of your Process window. You can optionally center the window, or use the default position that is set by the template wizard.

Report Customization

The settings here control the appearance of the Report procedure generated by the Report Wizard. These settings can be saved for use in other future applications.

Procedure
Procedure Name
Enter a name that the Report Wizard will use to generate procedure names in your application. The template macro symbol, %FileName, is required, and extracts the Dictionary file name for each file selected by the wizard. You can modify this line with other template macros and text if you wish. No spaces are allowed in procedure names. This name is used for files that have no keys defined

Procedure "With Key"
This prompt is similar to Procedure Name, but is used for designated files that have a key (or keys) defined in the Dictionary Editor.

Progress Window

Caption
Text that is entered on this line will be used as the description of the report's progress window, and appear in the progress window's title bar.

Background
Enter a default image here to use as a graphic or watermark for your report's progress window.

Mode
If you have designated a background image to use for your report's progress window, this option becomes available to control if the image is tiled, stretched, or centered.

Icon
Enter a default icon, or press the ellipsis button to select an icon file for use in your report's progress window. This will allow your window to be minimized if needed.

Font
Press the Font button to select a default font to use for the report's progress window. Sample text shown below the button is provided to allow you to review your selection.

Width
Enter the default width in dialog units for the report's progress window.

Height
Enter the default height in dialog units for the report's progress window.

Position
Select from the drop list to designate the initial position of your Progress window. You can center the window, or use the default position that is set by the template wizard.

Controls
There are four default controls in the Progress Window that you can customize: Progress, Pct Text, User String, Cancel Button.

Report Layout

General

The settings in this tab control refer to the settings of the current active report layout, whose name is displayed at the top.

Job Name
Enter the print job name to use for the Windows Print Manager. If omitted, the REPORT's label is used. The default setting is "Report %FileName".

Paper Type
Select the paper size for the report output from the drop list provided.

Paper Width
If you select "Other" as the Paper Type, you must enter a custom paper width.

Paper Height
If you select "Other" as the Paper Type, you must enter a custom paper height.

Margins
Margin setting control the printable area of your reports. Specify the Top, Bottom, Left and Right margin settings in thousandths of an inch

Check the **Automatic Adjust Top and Bottom** box to allow auto resizing of the margins when the report header or footer heights are resized.

Press the **Adjust Using Header and Footer Size** button to manually update the margin settings to conform to any changes made to the header and footer height positions. This button is only enabled if the Automatic Adjust Top and Bottom box is unchecked, to avoid sizing conflicts.

Orientation
Specify here what paper orientation the report layout will use (Portrait or Landscape).

Font
Press the Font Button to specify a default font to use for reports that the wizard will generate. There are several other places where you can override this setting.

Show Preview
Check this box to designate that all reports generated by the wizard will have a Print Preview window associated with it.

Preview Maximized
If the Show Preview box is checked, you can designate here that the Print Preview window will open in maximized mode (full screen).

Initial Zoom Setting
If the Show Preview box is checked, you can designate an initial zoom setting from this setting.

Header

The HEADER structure declares the output that prints at the beginning of each page or group.

Add Header
Check this box to allow your reports that are generated by the wizards to declare a HEADER section.

Background Color
If your user has support for color printing, you can designate a color to use as a background for the report's header.

Report Title
Enter a string or expression that will be used as the report title for each report generated by the report wizard. Since you can generate many report from different tables specified, it is a good idea to use a template macro here. The default is "Report %FileName file"

Position Y
Specify the starting position of the report title, relative to the start of the printable header area. Value is expressed in thousandths of an inch.

Justification
Specify how the contents of header text will be justified. The default is Center, but you can also specify Left or Right justified.

Title Font
The report title's font can be controlled with this setting. Press the Title Font button to designate the appearance of the title text.

Position
Enter the starting X and Y coordinates for the upper left corner of the header area, which is measured in thousandths of an inch, and relative to the upper left corner of the report's printable area. Enter the header area's width and height, also measured in thousandths of an inch

Font
Press the Font button to select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for all controls in the header section. A sample of the selected font is displayed on the tab control.

Header Box
Check the Add Header box to add a box control to the report header. You will then be able to specify a color, Top (or width), and Height settings.

Detail

The detail area is the "body" of the report. Usually, your most important and relevant data will be printed here.

Background Color
If your user has support for color printing, press the ellipsis button to designate a color here to use as a background for the report's detail area.

Position
Enter the starting X and Y coordinates for the upper left corner of the detail area, which is measured in thousandths of an inch, and relative to the upper left corner of the report's printable area, or to the last item printed in the detail print area. Enter the detail area's width and height, also measured in thousandths of an inch.

Think of this setting as the "record" or "row" area of your report.

Font
Press the Font button to select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for all controls in the Detail section. A sample of the selected font is displayed on the tab control.

Footer

The report FOOTER area is normally used to print text and data at the end of each page.

Add Footer
Check this box to allow your reports that are generated by the wizards to declare a FOOTER section.

Background Color
If your user has support for color printing, press the ellipsis button to designate a color here to use as a background for the report's footer area.

Position
Enter the starting X and Y coordinates for the upper left corner of the footer area, which is measured in thousandths of an inch, and relative to the lower left corner of the report's detail area, or to the last item printed in the detail print area. Enter the detail area's width and height, also measured in thousandths of an inch.

Font
Press the Font button to select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for all controls in the report's footer section. A sample of the selected font is displayed on the tab control.

Form

The Form Band is normally used to specify constant text or graphics that print on every page (for example, a watermark or a tax form)

Add Form
Check this box to allow your reports generated by the wizards to specify a Form section.

Background Color
If your user has support for color printing, press the ellipsis button to designate a color here to use as a background for the report's form band area.

Margins
Margin settings control the printable area of your form band. Specify the Top, Bottom, Left and Right margin settings in thousandths of an inch

Font
Press the Font button to select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for all controls in the report's Form Band section. A sample of the selected font is displayed on the tab control.

Image:

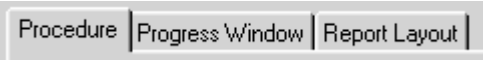
Add Image
Check this box to allow an image control to be populated on the Form band. This check box also enables the remaining options.

Image File
Press the ellipsis button to select the name of the image file to be used in the populated image control. Any image format supported by the Windows print engine may be selected.
Mode

Select from the drop list how the image file will be rendered on the report. Select from tiled, stretched, and centered.

Image Position
If width and height are zero(0), the image will use the settings specified by the form band. Set the X and Y position to designate the anchor position of the top left corner.

Label Customization



The settings here control the appearance of the Report procedure with a Label layout generated by the Label Wizard. These settings can be saved for use in other future applications.

Procedure

Procedure Name
Enter a name that the Label Wizard will use to generate procedure names in your application. The template macro symbol, %FileName, is required, and extracts the Dictionary file name for each file selected by the wizard. You can modify this line with other template macros and text if you wish. No spaces are allowed in procedure names. This name is used for files that have no keys defined

Proc. "With Key" Name
This prompt is similar to Procedure Name, but is used for designated files that have a key (or keys) defined in the Dictionary Editor.

Progress Window

Name
Text that is entered on this line will be used as the description of the report's progress window, and appear in the progress window's title bar.

Background
Enter a default image here to use as a graphic or watermark for your report's progress window.

Mode
If you have designated a background image to use for your report's progress window, this option becomes available to control if the image is tiled, stretched, or centered.

Icon
Enter a default icon, or press the ellipsis button to select an icon file for use in your report's progress window. This will allow your window to be minimized if needed.

Font
Press the Font button to select a default font to use for the report's progress window. Sample text shown below the button is provided to allow you to review your selection.

Width
Enter the default width in dialog units for the report's progress window.

Height
Enter the default height in dialog units for the report's progress window.

Position
Select from the drop list to designate the initial position of your Progress window. You can center the window, or use the default position that is set by the template wizard.

Controls
There are four default controls in the Progress Window that you can customize:
Progress

Report Layout

General
The settings in this tab control refer to the settings of the current active report layout, whose name is displayed at the top.

Job Name
Enter the print job name to use for the Windows Print Manager. If omitted, the REPORT's *label* is used. The default setting is "%FileName Report".

Orientation
Specify here what paper orientation the report layout will use (Portrait or Landscape).

Font
Press the Font Button to specify a default font to use for reports that the wizard will generate. There are several other places where you can override this setting.

Show Preview
Check this box to designate that all reports generated by the wizard will have a Print Preview window associated with it.

Preview Maximized
If the Show Preview box is checked, you can designate here that the Print Preview window will open in maximized mode (full screen).

Initial Zoom Setting
If the Show Preview box is checked, you can designate an initial zoom setting from this setting.

Background Color
If your printer has support for color printing, you can designate a color to use as a background for the printed labels.

Font
Press the *Font* button to select the font (typeface), size, style (such as bold or italic), color, and font effects (underline and strikeout) for all controls in the label section. A sample of the selected font is displayed on the tab control**.**

Label Size

In most cases, the report label settings, based on the Label Group and Label Type, will be preset for you. Use this tab control's settings if you have specified a Custom Label Group or Custom Label Type. The graphic provided on this tab control guides you through the parameters available here.

All measurements are in thousandths of an inch by default.

Width
Enter the width of the individual label's printable area.

Height
Enter the width of the individual label's printable area.

Top margin
Enter the height of the label's non-printable area, measured from the top of the page. For example, if you want the first label to print one inch from the top of the page, enter 1000.

Left margin
Enter the height of the label's non-printable area, measured from the left edge of the page. For example, if you want the first label to print one-half inch from the left edge, enter 500.

Number
Enter the number of labels to print across the page.

Number down
Enter the number of labels to print down the page.

Horizontal
Enter the horizontal distance between labels, measured from the left edge of one label to the left edge of the next label.

Vertical Pitch
Enter the vertical distance between labels, measured from the left edge of one label to the left edge of the next label.

Paper Type

Select a paper size from the drop list corresponding to the page size that you will print labels on. If you select Other, you can customize the page size to any length and height, using the Paper Width and Paper Height settings.

Window Customization

The settings here control the appearance of the Report procedure with a Label layout generated by the Label Wizard. These settings can be saved for use in other future applications.

Window

Name
Enter a name to use as the default caption, or title, for all windows generated.

Background
Enter a default image here to use as a graphic or watermark for your Window procedure control area.

Mode
If you have designated a background image to use for your Window procedure, this option becomes available to control if the displayed image is tiled, stretched, or centered.

Icon
Enter a default icon to use for your Window procedure. This will allow your window to be minimized if needed.

Font
Press the Font button to select a default font to use for the Window procedure. Sample text shown below the button is provided to allow you to review your selection.

Position
Select from the drop list to designate the initial position of your window. You can optionally center the window, or use the default position that is set by the template wizard.

In addition, click on the **System Menu** check box to add the Windows System Menu to your Window procedure. If you will be using any special entry control pictures on the window, you can also click on the **Entry Patterns** check box to allow the display of special formatting picture information (Example: phone numbers or date pictures)

Buttons

The Window Wizard has three optional buttons that you can choose to populate. They are the **OK**, **Cancel** and **Help** button controls. Check on each "Enable *name* button" box to activate the appropriate button dialogs.

Date and Time

Add Date and Time
Check this box to include an automatic Report Date and Time template control on your reports generated by the wizards.

Position
Specify which report band to populate the Date and Time controls. Select from *Header*, *Footer*, *Detail*, *Form*, and **both** *Header and Footer*

Date and Time Pictures
Press the ellipsis button on the appropriate Date or Time entry to call up the *Edit Picture* dialog window, and build a picture token to use for each one. The default picture is *@D17* and *@T7* respectively.

Date and Time Positions
Specify the exact position within the selected bands where the date and time controls and prompts will be populated.

Page Number

Add Page Number
Check this box to include an automatic Page Number template control on your report.

Position
Specify which report band to populate the Page Number control. Select from *Header*, *Footer*, *Detail*, *Form*, and **both** *Header and Footer*

Page Picture
Press the ellipsis button to call up the *Edit Picture* dialog window, and build a picture token to use for the page number. The default picture is *@pPage <<#p*

Pos X and Pos Y
Specify the exact position within the selected bands where the control will be populated.

Save As Theme Dialog

This dialog is available at the end of each Theme Wizard. The following options are available:

Create a new theme?

Check this box to write your current changes to a new theme name and associated external file.

Save the choices that you made?

Check this box to preserve your settings on this window. For example, if you always want to save the report layout with the theme, check this box.

Theme Name

Enter a name that will be used in the Theme Selection Drop List for all future wizard use.

File Name

Enter the physical file name that will be used to write your current changes to. Although it is not required, it is probably a good idea to keep the file name consistent with the Theme name entered above.

Change Layout Name?

Check this box to write the current report layout changes to a new layout name.

New Layout Name

Enter the new layout name here.

Save It

Press this button to write the current changes to the theme file name entered above.

Press the **OK** button to maintain your current checkbox settings on this window, or press the **Cancel** button to reset them to their original values when you entered this window.

Save Theme Dialog

Save the choices that you made?

Check this box to preserve your settings on this window. For example, if you always want to save the report layout with the theme, check this box.

Change Layout Name?

Check this box to allow a new layout name to be saved. If you leave this box unchecked, the current report layout name will be saved with any changes you have made.

Save It

Press this button to write the current changes to the *Default.TFT* theme file.

Press the **OK** button to maintain your current checkbox settings on this window, or press the **Cancel** button to reset them to their original values.

Theme Design Dialog

Wizard Prompts

The Prompts buttons allow default settings to be established for the selected theme. Each prompt is described below.

Application

Control Model
There are three models the wizard can use to create applications: Button, Toolbar, or Both.

Button
The wizard builds the application with traditional Insert, Change, Delete, OK, and Cancel command buttons that appear on each dialog.

Toolbar
The wizard builds the application with global toolbar command buttons that appear on the application frame. The toolbar buttons control each dialog. See Control Templates-FrameBrowseControl for more information.

Both
The wizard builds the application with both the traditional dialog command buttons and the global toolbar command buttons.

Reports
Check the **Generate Reports for each file** box to automatically generate report procedures for this selected theme.

Browse

Call Update Procedure
Check this box to allow this theme to always generate a Form (Update) procedure for each Browse that is generated.

Child Files
If the browse wizard is accessing a parent table, check this box to allow a button to be generated for each child file defined in the Data Dictionary.

Parent File Handling
If the browse wizard is accessing a child table, you can specify how the browse will process the records:

Do not select by parent record
Do not limit the browse – browse all records.

Select parent record via button
Browse only the child records for a specific parent record. Provide a button to select the parent record.

Assume that the parent record is active
Browse only the child records for a specific parent record. Assume the parent record is already active, and do not provide a button.

Select Button
Check this box to provide a "Select" button that displays when the Browse procedure is called to select a record, but is hidden when the Browse is called to update records.

Form

Updating Records
Enable the appropriate check boxes to allow the wizard to include a Save Button extension to any Form that uses the selected theme.

Messages
See the Save Button Control Template for more information regarding messages.

Record Validation
Enable the appropriate check boxes to allow the wizard to include the Record Validation extension to any Form that uses the selected theme.

Child File Handling
If the primary table to be updated by the Form is defined as the parent in a parent/child relationship defined in the Data Dictionary, you can specify this theme to handle the child file in one of the following ways:

Place children on tabs
Access children on push buttons
Do not provide child access

Report

Columns
How many columns do you want the report to use? Type the number of columns for your report. The Report Wizard distributes the report columns evenly across the columns.

Theme Maintenance Wizard

The Theme Maintenance Wizard is a powerful template utility that allows you to Add, Modify, or Delete selected template wizard themes.

Themes are defined as text files that control a wizard's generated output and, in addition, the wizard's default prompt settings that are presented to the developer. A theme file has a default file extension of TFT, and is stored in the TFT sub folder of the Clarion TEMPLATE folder. This path, and the list of existing themes, is found in the C6TFT.INI file located in the Clarion BIN folder.

The Theme Maintenance Wizard contains the following options:

Theme Selection

Theme From the drop down list, select an existing theme that you wish to add, modify, or delete. If you are adding a new theme, you will need to select an existing theme. After you make modification to the selected theme, you can use the Save As option.

Report Layout From the drop down list, select an existing report layout that you wish to add, modify, or delete. Changes to the Report Wizard will be applied to this layout.

Operations

The buttons shown on this wizard sheet direct you to one of four operations. You can **Design Theme**, based on the theme selected on the previous window. After your modifications, return to this window to **Save Theme Default.TFT** (the selected theme), or **Save As** to a new theme name.

You can also remove the selected theme permanently by pressing the **Delete Theme Default.TFT** button.

XML Support Templates

Export to XML code template

The Export to XML code template is used to write the contents of a valid data source (or origin) to an external XML file. This code template is actually a wrapper around the built-in class library support for XML parsing. The following prompts are provided:

Data Origin Press the ellipsis button to select a valid label of a GROUP, FILE, VIEW or QUEUE that holds the XML contents to export to an XML file.

Mapping File Press the ellipsis button to select a valid XML mapping file to use during the export process. In Clarion terms, mapping refers to associating an XML tag or schema to a database column name. If you wish to select this file at runtime, check the **Select mapping file at runtime** check box.

Export File Press the ellipsis button to select the valid XML source file name to export to. If you wish to select this file at runtime, check the **Select export file at runtime** check box.

XML Style Select *Tag-based* or *Attribute-based*. This setting controls the XML formatted output style.

Example – Tag based output:

```
<?xml version="1.0" ?>
- <dataroot>
- <Table>
  <id>1</id>
  <firstname>Fred</firstname>
  <lastname>Flintstone</lastname>
  <gender>M</gender>
  </Table>
```

Example – Attribute based output:

```
<?xml version="1.0" ?>
- <dataroot>
  <row id="1" firstname="Fred" lastname="Flintstone" gender="M" />
  <row id="2" firstname="Andrew" lastname="Guidroz" gender="M" />
  <row id="4" firstname="Gavin" lastname="Holiday" gender="M" />
  <row id="35" firstname="Hello" lastname="There" gender="f" />
  </dataroot>
```

For more information, please review the *Clarion XML Support Reference Guide* PDF file included in your install.

Import From XML code template

The Import From XML code template is used to read the contents of an external XML into a valid target destination. This code template is actually a wrapper around the built-in class library support for XML parsing.

The following prompts are provided:

Import File Press the ellipsis button to select a valid XML source file to import. If you wish to select this file name at runtime, check the **Select import file at runtime** check box.

Mapping File Press the ellipsis button to select a valid XML mapping file to use during import. In Clarion terms, mapping refers to associating an XML tag or schema to a database column name. If you wish to select this file at runtime, check the **Select mapping file at runtime** check box.

Destination Press the ellipsis button to select a valid label of a GROUP, FILE, VIEW or QUEUE to hold the contents of the import XML file.

For more information, please review the *Clarion XML Support Reference Guide* PDF file included in your install.

Index: