Help Contents

Select an item from the list below. If Windows Help is new to you, choose Help|How to Use Help.

Overview of the dBASE Component Builder

Conversion Operations

Preferences

The Conversion Process

The conversion process of the Component Builder is simple and straightforward. The dBASE IV files listed below on the left convert to the dBASE for Windows files listed on the right. Each of the file types listed below appears as an option on the File menu.

dBASE IV File	dBASE for Windows File	
.PRG	.MNU	(Extracts only menu-related code)
.PRG	.WFM	(Extracts only @...SAY/GET code)
.FMT	.WFM	(Extracts screen controls and layout)
.FRM	.RPT	(Extracts report fields and layout)
.LBL	.RPL	(Extracts fields and layout)

The Component Builder converts files by searching for certain dBASE IV commands and translating them into the corresponding dBASE for Windows commands. The table below summarizes the dBASE IV and corresponding dBASE for Windows commands.

dBASE IV Command	dBASE for Windows Command
DEFINE MENU	CLASS <menuName>
DEFINE PAD	DEFINE MENU <padName.menuName>
DEFINE BAR	DEFINE MENU <barName.padName.menuName>
ON SELECTION	<barName.padName.menuName.onClick>
@...SAY	DEFINE TEXT
@...GET	DEFINE ENTRYFIELD
@...TO	DEFINE RECTANGLE
@...FILL	DEFINE RECTANGLE <with color>

.PRG Files to .MNU Files

Updating a dBASE IV menu structure is simple and straightforward. dBASE IV .PRG files that contain DEFINE MENU commands and structures can be individually selected to generate new .MNU files based on a marked selection in the Component Builder selection text editor.

The .PRG to .MNU option is an interactive process that makes it possible to select many different menu structures (one at a time) and submit the marked text selections to the Component Builder for processing. When selecting menu structures, it is not mandatory that you select only dBASE IV menu commands. If there are non-menu commands in the selection, the Component Builder ignores them when generating the new .MNU file.

After you choose the .PRG to .MNU option and pick a .PRG file to work with, the selection text editor lets you interactively mark specific menu code sections from the .PRG file. Once you have marked a menu structure, press the BUILD button on the text editor form. The Component Builder prompts you for an .MNU file name in which to save the output. Control returns to the selection text editor once the .MNU file has been created.

The Component Builder assumes that menu commands occur in the following general sequence:

```
DEFINE MENU ...
DEFINE PAD ...
DEFINE PAD ...
DEFINE PAD ...

DEFINE POPUP ...
DEFINE BAR 01 ...
DEFINE BAR 02 ...
DEFINE BAR 03 ...

ON PAD ...
```

The .MNU file that is generated contains code that replicates the original dBASE IV-style menu, but in dBASE for Windows style. The .MNU file contains the structure of the menu but does not contain the actions assigned to each menu item. To assign actions, use the menu item OnClick property. To use the .MNU file with a dBASE for Windows form, enter the .MNU file name in the form's MenuFile property.

The new menu code is also compliant with the dBASE for Windows Menu Designer, which you can use to enhance it further.

.PRG Files to .WFM Files

Updating @...SAY/GET code in a .PRG file is similar to updating menu code. dBASE IV .PRG files that contain @...SAY/GET commands can be individually selected to generate new .WFM files based on a marked selection in the Component Builder selection text editor.

The .PRG to .WFM option is an interactive process that makes it possible to select many different @...SAY/GET command lines and submit the marked text selections to the Component Builder for processing. When selecting code, it is not mandatory that you select only @...SAY/GET commands. If there are other commands in the selection, the Component Builder ignores them when generating the new .WFM file.

After you choose the .PRG to .WFM option and pick a .PRG file to work with, the selection text editor lets you interactively mark specific code sections from the .PRG file. Once you have marked some code, press the BUILD button on the text editor form. The Component Builder prompts you for a .WFM file name in which to save the output. Control returns to the selection text editor once the .WFM file has been created.

The .WFM file that is generated contains code that replicates the original @...SAY/GET commands, but in dBASE for Windows style. The new code is compliant with the dBASE for Windows Form Designer, which you can use to enhance it further. For example, if the @...SAY/GET commands display memory variables, you need to add code for initializing the variables to the OnOpen procedure, or initialize the Value property of each entry field.

.FMT Files to .WFM Files

Updating dBASE IV or dBASE III PLUS screen format files to the new Windows style format requires little effort. If you have existing .FMT files that contain @...SAY...GET commands, select an individual .FMT file or place a collection of .FMT files in a new catalog. The Component Builder generates new files of the same name with a .WFM extension.

Each .WFM file contains code that replicates the original dBASE IV format file, but in dBASE for Windows coding style. The new code is compliant with the dBASE for Windows Form Designer, which you can use to enhance it further.

.FRM Files to .RPT Files

Updating dBASE IV report form files to the new Windows format requires little effort. If you have existing .FRM files based on existing database files, select the .FRM file you want. The Component Builder generates a new file of the same name with an .RPT extension.

The .RPT file conforms to the new Crystal Reports format and can be loaded into Crystal Reports for dBASE for additional modification or enhancement.

When updating a dBASE report form file (.FRM), the Component Builder prompts you for the name of the database or query file that is associated with the report form file. This entry is required; without a view file, Crystal Reports cannot load the new .RPT file.

.LBL Files to .RPL Files

Updating dBASE IV label files to the new Windows format is almost indentical to the conversion of .FRM files.

If you have .LBL files based on database (table) or query files, select the .LBL file you want. The Component Builder generates a new file of the same name with an .RPL extension.

You are prompted for the name of a database or query view file that is associated with the label form file. This entry is required; without a view file, Crystal Reports cannot load the new .RPL file.

The .RPL file conforms to the new Crystal Reports label format and can then be loaded into Crystal Reports for dBASE for additional enhancement.

Files That Can Be Converted

In the Windows environment the approach to elements such as the user interface is different from the approach in the DOS environment. It is unrealistic to expect 100% compatibility and still take advantage of a graphical user interface. With the help of the Component Builder, you can use existing menus, forms, reports and labels and begin to exploit some of the advantages that Windows offers.

dBASE IV files that need conversion to take advantage of the object capabilities include menus, forms, and window-oriented code as well as reports and labels. The Component Builder updates .PRG, .FMT, .FRM, and .LBL files to their respective file formats under dBASE for Windows.

The File menu of the Component Builder lists options for updating each type of file. Additionally, a Catalog option makes it easy to batch multiple files together in a separate catalog file, so you can submit just the catalog file for conversion. This makes it possible to update entire applications with a single Component Builder menu choice. Catalog files submitted for conversion can contain any dBASE file types, although only the .FRM, .LBL, and .FMT files are processed. Menu conversions are only handled interactively.

Conversion Guidelines

Guidelines

Introduction to the Conversion Process

The purpose of this utility is to bring your DOS resources into Windows without having to re-create them. Menus, forms, reports, and labels all share some fundamental qualities and definitions that can be translated into dBASE for Windows objects.

To make the conversion process as easy as possible, the Component Builder supports single file selection and wildcard file selection. To learn how to best use this product, experiment with a few source files that need translation. Once you get a feel for the way the utility works, you can then work on a larger set of files.

Typically, dBASE applications comprise many forms, labels, and reports, but in most cases, there is one menu structure. A good place to start is the conversion of a dBASE IV menu structure.

Before conversion, make backup copies of the files you intend to update. Place them in a separate subdirectory, and use these copies for conversion instead of the original files. By default, the conversion will place the new files in that same directory.

A Recommended Approach

A few steps can make the conversion process work smoothly.

1. Create a new directory to store all the source and converted files.
2. Copy all the files that you want to update into this new directory. (This makes it easy to select files for conversion and creates a set of backup files of the original application.)
3. Start dBASE for Windows. Create a catalog file that contains all the .FMT, .LBL, and .FRM files you want to update. (You can drag-and-drop files from the Navigator to the Catalog window.)
4. Start the Component Builder from the Navigator by changing the Current Directory to the dBASE for Windows UTILITY directory. Choose the Programs file type, then run CB.PRO.
5. From the Component Builder File menu, select the Catalog command and pick the catalog that you created in step 3. The Component Builder updates each file in the catalog sequentially.
6. When the conversion process has finished, you can choose to view the generated code with a text editor. The menus, forms, and reports are also ready to be further customized with the dBASE for Windows Form Designer and Menu Designer.

Additional Support

Component Builder source code can be downloaded through the Borland Online Services. Please check the Online Services section of the Borland Assist Support and Services Guide included in the package. The file name is CB.ZIP.

Code-Generation Options Dialog Box

Use this dialog box to set how code is generated by the Component Builder. Choose Preferences|Code-Generation Options to display this dialog box.

Dialog Box Options

Prompt For View File [Catalog Update]

When you select a single .FMT file, the Component Builder prompts for the view, a database file (table) or query file, upon which the .FMT file is based. The file you select is later assigned to the View property of the new form.

When you submit a batch of .FMT files, you can suppress the view file prompt so that updating does not stop each time an .FMT file is encountered. When you uncheck this check box, the Component Builder updates files in the catalog non-stop. Afterward, you need to fill in the View property for each updated form. You can do so using the Object Inspector in the Form Designer.

Save To Local Directory

This option specifies the directory in which newly created files are saved. If you uncheck this option, files are saved to the same directory they are read from. If you check this option, files are saved to the local directory. The local directory is the directory specified in File|Change Directory. By default, the local directory is the UTILITY directory.

Include Source Comments In Forms

Source comments are programming notes that are automatically generated by the Component Builder. This option lets you include comments about the source code being generated.

Logicals to Check Boxes

This option lets you create a check box when a logical field or variable entry is encountered in an .FMT file.

Indent Level

This option is a spin box that lets you increment or decrement the indentation level of the source being generated. Values for this option range from a minimum of a 1-character indent level to a maximum of 9. The default value is 3.

Glossary

@...GET
@...SAY
CLASS
DEFINE RECTANGLE
DEFINE ENTRYFIELD
DEFINE MENU
DEFINE TEXT
@...FILL
.FMT
.FRM
Indent Level
.LBL
.MNU
.PRG
.RPL
.RPT
Source Comments
text
.WFM

.PRG

Source code that contains menu commands and structures. The Component Builder absorbs the menu structures contained in any program files based on a highlighted selection. All other source code in the selection is ignored, and only menu definitions are extracted and converted into .MNU formats for dBASE for Windows.

.FMT

Format files created for dBASE IV or dBASE III PLUS. These files can include @...GET, @...FILL, and @...TO statements. Color settings are also utilized.

.FRM

Report form files created with dBASE IV.

.LBL

Label form files created with dBASE IV.

.MNU

.MNU files contain DEFINE MENU commands that create menu objects for dBASE for Windows. These files are automatically generated by the Component Builder from existing .PRG files. .MNU files can also be created with the dBASE for Windows Form Designer.

.WFM

.WFM files are Windows form files generated as a result of submitting files to the Component Builder. .WFM files generally include a form definition as well as screen objects and controls converted from the source .FMT files. .WFM files are also created using the dBASE for Windows Form Designer.

.RPT

.RPT files are report form definitions generated by the Component Builder from DOS-based .FRM and .LBL files. .RPT files are specifically associated with Crystal Reports for dBASE.

.RPL

.RPL files are label form definitions generated by the Component Builder from DOS-based .LBL files. .RPL files are specifically associated with Crystal Reports for dBASE.

Source Comments

Source comments are programming notes that are automatically generated by the Component Builder.

Indent Level

A value that describes how many characters the source code is indented from the left margin or from each indent margin when generating source code with Component Builder. By default, this value is set at 3 characters but can be changed by choosing Preferences|Code-Generation Options.

CLASS
The dBASE for Windows command used to create a new class of object.

DEFINE MENU
The dBASE for Windows command used to create a menu object.

DEFINE TEXT
The dBASE for Windows command used to create a text object.

DEFINE ENTRYFIELD
The dBASE for Windows command used to create a data entry field.

DEFINE RECTANGLE
The dBASE for Windows command used to create a box object.

@...SAY

The dBASE IV command to display text onscreen.

Text Object

A text object is to dBASE for Windows what a SAY statement is to dBASE IV. The command DEFINE TEXT is used in dBASE for Windows to display text in a form.

@...GET
The dBASE IV command to prompt for user input onscreen.

@...FILL
The dBASE IV command to create a non-transparent box.