POWERSOFT
ENTERPRISE
SERIES

Advanced PowerBuilder Utilities

VERSION 4.0

PowerBuilder

December 1994

Contents

About This Manual

Subject

This manual describes how to use the Advanced PowerBuilder Utilities, a set of analysis and development tools that you can use to enhance quality and increase productivity.

Audience

This manual is for experienced PowerEuilder developers. It assumes that:

- You are comfortable using Microsoft Windows applications
- You are currently developing applications using PowerBuilder and understand the concepts and techniques described in the PowerBuilder *Building Applications* manual
- You understand SQL and how to use your site-specific DBMS

CHAPTER 1

Overview and Installation

About this chapter

This chapter gives you an overview of the Advanced PowerBuilder Utilities and provides installation instructions.

Contents

Advanced PowerBuilder Utilities overview

The Advanced PowerBuilder Utilities are a series of tools that extend the PowerBuilder development environment. The utilities automate, and in some cases eliminate, the traditional tasks involved in application development.

These tools include analysis utilities and development utilities.

This manual describes each utility, including its function, background, and the steps necessary to use the utility.

Analysis utilities

Tool	Description
Cross Reference	Examines the relationships between the objects that make up a PowerBuilder application
DataWindow Extended Attribute Synchronizer (DWEAS)	Updates attributes of existing DataWindow objects from the PowerBuilder extended attribute tables
DataWindow SQL Verifier	Checks the syntax of the SQL used by existing DataWindow objects
PowerBuilder Extended Attribute Reporter (PEAR)	Generates reports on the contents of the PowerBuilder extended attribute tables

Development utilities

Tool	Description
Image Editor	Creates icons, bitmaps, and cursors
Stored Procedure Update	Creates PowerScript statements that use stored procedures to update the database through a DataWindow
Install Diskette Builder	Creates install diskette sets for your PowerBuilder applications

Installation

❖ To install the Advanced PowerBuilder Utilities:

1 Insert the Advanced PowerBuilder Utilities diskette into your floppy drive.

2 Run the Advanced PowerBuilder Utilities Setup program in either of two ways.

- ♦ At the DOS prompt, type:

```
WIN A:SETUP
```

- ♦ Select File➤Run from the Windows Program Manager menu bar and type:

```
A:SETUP
```

3 Respond to the Setup prompts to name the directory where the files will be installed.

> ***Caution***
> *Install the Advanced PowerBuilder Utilities into your PowerBuilder directory.*

4 Browse the Read Me file while the install completes. This file contains last-minute documentation issues and other useful information.

CHAPTER 2

Cross Reference

About this chapter

This chapter describes the Cross Reference tool.

Contents

Cross Reference overview

The Cross Reference tool examines the relationships among objects that make up a PowerBuilder application. These objects include windows, menus, DataWindows, user objects, global variables, shared variables, instance variables, global functions, window functions, user object functions, and PowerScript functions. After running the Cross Reference process, you can analyze the relationships among these objects, sorting, saving to a database, or printing, as necessary.

About Cross Reference

It is easy to see what objects make up an application using the Library painter, but it may be difficult to remember where each object was used in the application. Cross Reference shows these relationships by providing a list of all of the objects in the application and the relationships that they have to each other. You can also display a list of the objects that are not referenced in the application.

With this view of the application, you can determine whether objects are being used effectively and if there are new objects that can be created to reduce duplication of effort. For example, using Cross Reference, you may detect that only one object is using a global variable, thereby allowing you to remove it. You may also notice that several descendent window objects are using two variations of the same global function. Those functions may be able to be combined and placed into the ancestor window, resulting in better performance and less code maintenance. Cross Reference also generates information about where DataWindow objects are used, (as standard DataWindows, dropdown DataWindows, and nested reports), where external functions have been declared, and where global functions have been used in DataWindow objects.

You can save Cross Reference results in the Cross Reference database for use at a later time.

How Cross Reference works

Cross Reference works at an application level. First, you indicate the application to examine. Then you indicate which libraries within the application to examine.

Cross Reference uses a two-phase process to analyze your application.

- **Phase one** finds all of the objects that are contained in the selected application.
- **Phase two** looks for the relationships among the referenced objects. Cross Reference looks in variable and attribute declarations, as well as in your PowerScript statements in events and functions. Phase two creates a grid-style DataWindow to present a report of all of the objects that were found and the objects referenced by each one. You may change the order of this display and print the results or save the results to the Cross Reference database for further processing.

Using Cross Reference

This section describes how to use Cross Reference:

- Starting Cross Reference
- Exiting Cross Reference
- Cross Reference workspace
- Selecting an application
- The Cross Reference process
- Examining Cross Reference results
- Sorting Cross Reference results
- Changing column order
- Specifying Cross Reference options

Starting Cross Reference

❖ To start Cross Reference, do one of the following:

- Double-click the Cross Reference icon from the Windows Program Manager.
- Double-click XREF.EXE in the Windows File Manager.
- Select File➢Run from the Windows Program Manager menu bar and type:

  ***directory name*\xref**

The Cross Reference banner displays until initialization finishes and the main window appears.

Exiting Cross Reference

❖ To exit Cross Reference, do one of the following:

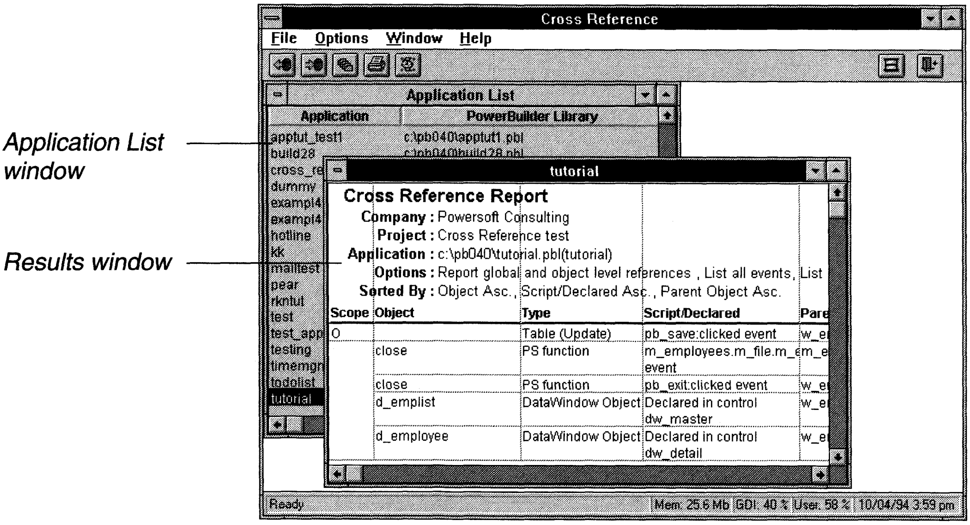
- Press ALT+F4.

- ♦ Select File➤Exit from the menu bar.
- ♦ Click the Exit button on the Cross Reference toolbar.
- ♦ Double-click the System menu in the top-left corner of the window.

Saving Cross Reference results
To save Cross Reference results, select File➤Save to Database from the menu bar before exiting from Cross Reference. When you exit from Cross Reference without saving, all of your unsaved results are deleted.

Cross Reference workspace

Here is an example of the Cross Reference workspace.



Application	PowerBuilder Library
apptut_test1	c:\pb040\apptut1.pbl
build28	c:\pb040\build28.pbl
cross_re	
dummy	
exampl4	
exampl4	
hotline	
kk	
mailtest	
pear	
rkntut	
test	
test_app	
testing	
timemgn	
todolist	
tutorial	

Cross Reference Report
Company : Powersoft Consulting
Project : Cross Reference test
Application : c:\pb040\tutorial.pbl(tutorial)
Options : Report global and object level references , List all events, List
Sorted By : Object Asc., Script/Declared Asc., Parent Object Asc.

Scope	Object	Type	Script/Declared	Pare
O		Table (Update)	pb_save:clicked event	w_e
	close	PS function	m_employees.m_file.m_e event	m_e
	close	PS function	pb_exit:clicked event	w_e
	d_emplist	DataWindow Object	Declared in control dw_master	w_e
	d_employee	DataWindow Object	Declared in control dw_detail	w_e

The Application List window lists the application objects and the PBL files in which they reside. Cross Reference displays a separate results window for each cross-referenced application.

If the Application List window is empty
Cross Reference uses the PB.INI file to obtain the application list. If the PB.INI file cannot be found, a message displays with an empty Application List window. This problem may indicate that the Advanced PowerBuilder Utilities were not installed in the PowerBuilder directory.

If a "Cannot read from drive" message displays
Cross Reference checks to see if each application named in the PB.INI file exists. If this application resides on a drive that is unavailable (such as an empty diskette drive or an unattached network drive), file errors will interrupt the process. Click Cancel to continue.

Selecting an application

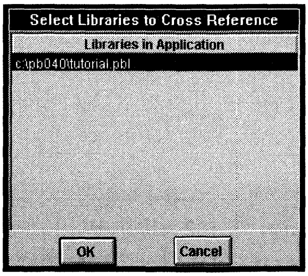
Once Cross Reference is running and you have specified the options you want, select one of your PowerBuilder applications from the Application List window. This window lists all of the PowerBuilder application objects and the PBLs in which each object resides, as defined in your PB.INI file.

❖ To select an application and begin the cross-reference process:

1 Do one of the following:

- ♦ Double-click on the application.
- ♦ Highlight an application row using the UP ARROW or DOWN ARROW keys, then press ENTER.

If there are multiple libraries in the application, the Select Libraries to Cross Reference dialog box displays. This dialog box allows you to selectively exclude libraries from the cross-reference process. If there is only one library in the application, Cross Reference begins analysis immediately.



c:\pb040\tutorial.pbl

2 Optionally exclude libraries by clicking on them.

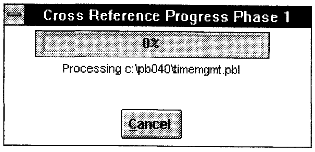
3 Click OK.

The Cross Reference process

Once you select an application object and the libraries to be cross-referenced, Cross Reference displays an empty results window, and begins analysis. The time that it takes to perform the analysis depends on the speed of your machine and network, and the number of objects in the selected application.

The Cross Reference process works in two phases:

- **Phase One** searches for all of the objects that are in the application. This process typically does not take long since it is only looking through libraries for objects. While Phase One is in progress, the following status window displays:



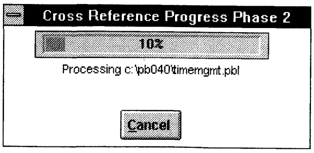
Processing c:\pb040\timemgmt.pbl

- **Phase Two** is a more intensive process and may take some time for a large application. In Phase Two, each object is converted to source format, parsed, and searched. Depending on the level of analysis you have specified for the Cross Reference, the parsing process views each object's scripts and attribute declarations for object references:

  - If you have selected the *Global Only* option, then Cross Reference searches for primary objects (that is, windows, DataWindows, menus, and global functions).

  - If you have selected the *Global and Object* option, then Cross Reference will also find for a given object the shared variables, instance variables, and object functions that are declared and referenced *within* the current object.

  While Phase Two is being performed, the following status window displays:

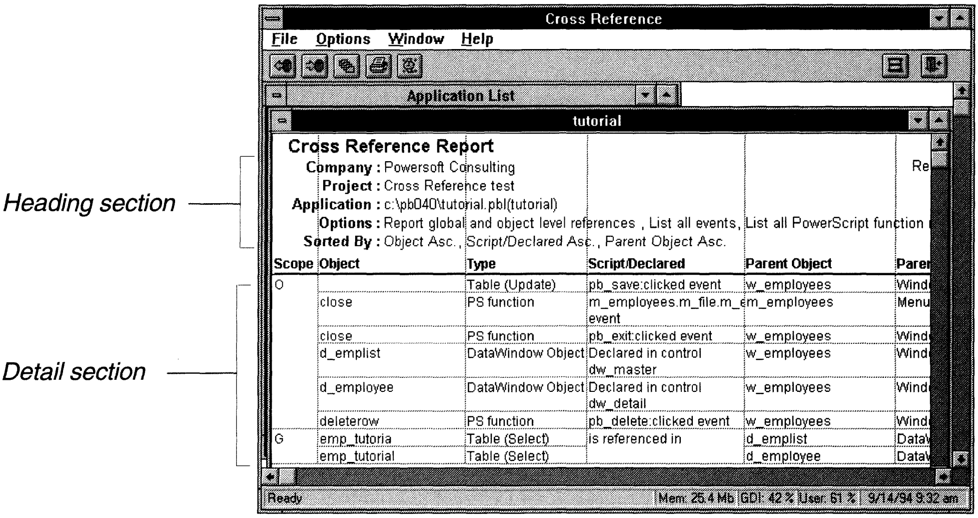


Processing c:\pb040\timemgmt.pbl

  Once Phase Two is finished, the Cross Reference results window displays.

Examining Cross Reference results

You can view the results of the Cross Reference analysis in the results window.



Cross Reference Report

Company : Powersoft Consulting
Project : Cross Reference test
Application : c:\pb040\tutorial.pbl(tutorial)
Options : Report global and object level references , List all events, List all PowerScript function
Sorted By : *Object Asc., Script/Declared Asc., Parent Object Asc.*

Re

Scope	Object	Type	Script/Declared	Parent Object	Parer
O		Table (Update)	pb_save:clicked event	w_employees	Wind
	close	PS function	m_employees.m_file.m_event	m_employees	Menu
	close	PS function	pb_exit:clicked event	w_employees	Wind
	d_emplist	DataWindow Object	Declared in control dw_master	w_employees	Wind
	d_employee	DataWindow Object	Declared in control dw_detail	w_employees	Wind
	deleterow	PS function	pb_delete:clicked event	w_employees	Wind
G	emp_tutoria	Table (Select)	is referenced in	d_emplist	Data
	emp_tutorial	Table (Select)		d_employee	Data

The title of the window containing the results of the Cross Reference depends on the name of the selected application. The Cross Reference report contains a heading section that describes the company, project name, application name, options selected, and sort order for the detail of the report.

The detail section of the report is made up of rows containing each referenced item. The rows are divided into these columns.

Column	Result
Scope	Location of the object with respect to the application. It can be one of two values: Global, meaning that the item is visible and accessible by the whole application, or Object, meaning that the item is a reference to an object or a component of an object.
Object	Name of the item referenced.
Type	PowerBuilder data type and its scope (global, public, instance, and so on) for Object.
Script/Declared	Name of the attribute or event script that references Object.

Column	Result
Parent Object	Name of the object that contains Script/Declared where Object is referenced.
Parent Object Type	PowerBuilder data type of Parent Object.
Parent Object Library	Filename of the library containing Parent Object.

At this point you can view, sort, print, or save the Cross Reference results, optionally modifying headings. You can also see a list of unreferenced objects and change the column order.

❖ To change the Company Name and/or Project Name in the report heading:

1 Select Options➤Set Options from the menu bar.

The Cross Reference Title Options dialog box displays.

2 Make your changes.

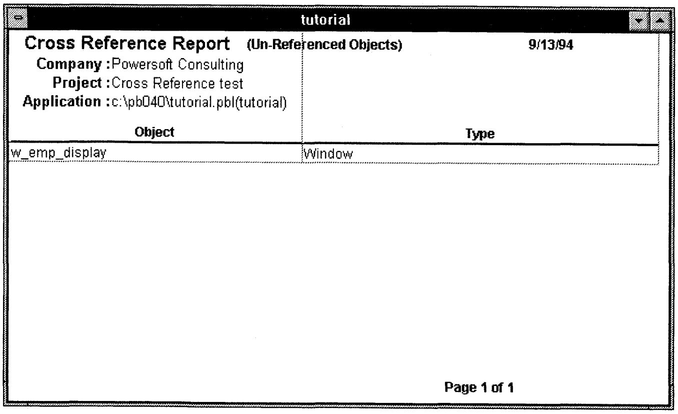
3 Click OK.

The changes are immediately reflected in the current report.

❖ To show a list of unreferenced objects:

♦ Select Options➤Show Unreferenced Object from the menu bar.

The results window displays with unreferenced objects.



Cross Reference Report (Un-Referenced Objects) 9/13/94

Company : Powersoft Consulting

Project : Cross Reference test

Application :c:\pb040\tutorial.pbl(tutorial)

Object	Type
w_emp_display	Window

Page 1 of 1

❖ To return to the original results window:

- ♦ Select Options➤Show Cross Reference Report from the menu bar.

  The original Cross Reference results window displays.

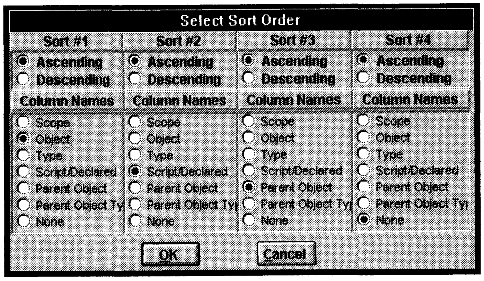
Sorting Cross Reference results

You can make the Cross Reference report more useful by specifying up to four levels of sorting for the results window.

❖ To sort a Cross Reference results window:

1 Select Options➤Sort from the menu bar.

The Select Sort Order dialog box displays.



Select Sort Order

Sort #1	Sort #2	Sort #3	Sort #4
● Ascending	● Ascending	● Ascending	● Ascending
○ Descending	○ Descending	○ Descending	○ Descending
Column Names	Column Names	Column Names	Column Names
○ Scope	○ Scope	○ Scope	○ Scope
● Object	○ Object	○ Object	○ Object
○ Type	○ Type	○ Type	○ Type
○ Script/Declared	● Script/Declared	○ Script/Declared	○ Script/Declared
○ Parent Object	○ Parent Object	● Parent Object	○ Parent Object
○ Parent Object Ty	○ Parent Object Ty	○ Parent Object Ty	○ Parent Object Ty
○ None	○ None	○ None	● None

You can choose up to four levels of sort order in which to display the Cross Reference results. The default is to sort the data by Object, then Script/Declared, and then Parent Object.

2 For each level of sorting, select the order (Ascending or Descending), and display column (Scope, Object, Type, Script/Declared, Parent Object, Parent Object Type, or None).

3 Click OK.

Cross Reference redisplays the results window using the sort criteria you specified.

Changing column order

In addition to modifying sort order, you can also display columns by referenced object or by parent object. By default, the Cross Reference Results window displays columns in order by referenced object.

❖ To display columns ordered by parent object:

- ♦ Select Options➤Report by Parent Object from the menu bar.

  Cross Reference moves the columns relating to parent object to the left side of the report. It also sorts the report by parent object and parent type.

❖ To display columns ordered by referenced object:

- ♦ Select Options➤Report by Referenced Object from the menu bar.

  Cross Reference moves the columns relating to the referenced object to the left side of the report. It also sorts the report by scope and object.

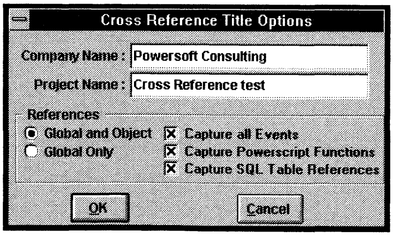
Specifying Cross Reference options

You can adjust default settings using the Cross Reference Title Options dialog box.

❖ To control Cross Reference options:

1 Select Options➤Set Options from the menu bar.

The Cross Reference Title Options dialog box displays.



Company Name : Powersoft Consulting

Project Name : Cross Reference test

References

● Global and Object ☒ Capture all Events

○ Global Only ☒ Capture Powerscript Functions

☒ Capture SQL Table References

2 Specify options, as necessary:

- Company Name or Project Name for the header of the report.
- Whether to cross-reference global and object references or just global references. If you select *Global and Object*, any objects that contain shared variables, instance variables, or object functions that are referenced from *within* the selected object are analyzed in the Cross Reference report.
- Whether to cross reference all events, PowerScript functions, and SQL table references.

3 Click OK.

Cross Reference menu bar

Cross Reference contains a menu bar and a toolbar. This section describes the menu bar and corresponding toolbar options in more detail.

File menu

The File menu contains choices that perform actions on Cross Reference windows.

Menu item	Activity	Shortcut key	Toolbar button
Load Report	Displays the Select an Application dialog box, which allows you to select a report from the Cross Reference database.		⇦
Save to Database	Saves the current report in the Cross Reference database.		⇨
Close	Closes the current Cross Reference report window. You cannot close the Application List window.	CTRL+F4	⊟
Print	Prints the current Cross Reference report or Application List window.	CTRL+P	🖨
Print Preview	Shows you what the Cross Reference report will look like when you print it. You can see whole pages one or two at a time, and you can see elements of the report that are not shown in normal view.		
Print Setup	Displays the standard Windows Printer Setup dialog box.		
Exit	Exits the Cross Reference tool.	ALT+F4	⇥

Options menu

The Options menu contains choices that allow you to change the default sort ordering for the display windows and to change the Cross Reference defaults.

Menu item	Activity	Shortcut key	Toolbar button
Sort	Displays the Select Sort Order dialog box, which allows you to sort Cross Reference results.	CTRL+S	▣
Set Options	Displays the Cross Reference Title Options dialog box, which allows you to set report header options and to specify the level of analysis performed by the Cross Reference process.		
Delete Application Info	Deletes the application from the Cross Reference database.		
Show Un-Referenced Objects	Displays a window listing objects that are not referenced in the application. You may be able to delete these objects if they are not used elsewhere. This choice is hidden when the unreferenced objects window is displayed.		▣
Show Cross Reference Report	Displays the Cross Reference window. This choice is only available when the Unreferenced Objects window is displayed.		▣
Report by Referenced Object	Displays the Cross Reference window, with columns relating to the referenced object displayed on the left. This is the default.		
Report by Parent Object	Displays the Cross Reference window, with columns relating to the parent object displayed on the left.		

Window menu

The Window menu contains choices for altering the placement of the windows within the MDI frame. It also allows you to control toolbar display.

Menu item	Activity
Tile Horizontal	Arranges the currently displayed windows side-by-side so that they do not overlap and so that each window takes up the same amount of space in the MDI client area.
Tile Vertical	Arranges the currently displayed windows one above the other so that they do not overlap and so that each window takes up the same amount of space in the MDI client area.
Layer	Enlarges the current window to fill the entire MDI client area, overlaying any other open windows.
Cascade	Causes all of the open windows to cascade from the upper left across to the lower right of the client area, one overlapping the other.
Arrange Icons	Arranges the icons of minimized windows within the MDI frame.
Tool Bars	Displays the Customize Toolbars dialog box, which allows you to control toolbar display.

In addition to the choices above, the Application List window and any open Cross Reference windows are displayed as items under the Window menu. You can activate any of the open windows by selecting its menu item.

Help menu

The Help menu contains choices for displaying online Help and the Cross Reference application's About dialog box.

Menu item	Activity
Overview	Displays online Help
Contents	Displays online Help
About	Displays the Cross Reference application's About dialog box, which provides identification and copyright information for the Cross Reference application

Cross Reference database

Cross Reference can save results to the XREF database. You can display previously saved results by selecting File➤Load Report from the menu bar. In addition, you can create customized reports that access the XREF database.

This section outlines tables and columns in the XREF database.

App_info table

This table contains Cross Reference report information for the application.

Primary key Application

Foreign key relationships App_info.application points to app_objects.application and xref_info.application

Field	Description
Application	Application name
Company	Company name used in report header
Project	Project name used in report header
Report_globals	Whether the report included Global and Object references or Global references only
list_sql	Whether the report included SQL table references
List_ps_func	Whether the report included PowerScript functions
List_all_events	Whether the report captured all events
Sort	String indicating sort order
Rep_date	Date the report was run
Rep_time	Time the report was run
Pbl	Filename of the PBL containing the application object

App_objects table

This table contains one row for every object in an application.

Primary key Application and name

Foreign key relationships App_info.application points to app_objects.application

Field	Description
Application	Application name
Name	Object name
Type	Object type
Library	Library containing the object

Xref_info table

This table contains cross reference information for all application objects.

Primary key Object_ref, object_ref_type, event, referenced_in, ref_in_type, application

Foreign key relationships App_info.application points to xref_info.application

Field	Description
Object_ref	Object name
Object_ref_type	Object type
Event	Event or control where object is used
Referenced_in	Object containing the event or control where the referenced object is used
Ref_in_type	Object type of the object named in the previous column
Pbl	Filename of the PBL containing the referenced object
Application	Name of the application containing the referenced object
Scope	Global or object.

CHAPTER 3

DataWindow Extended Attribute Synchronizer (DWEAS)

About this chapter

This chapter describes the DataWindow Extended Attribute Synchronizer (DWEAS).

Contents

DWEAS overview

The DataWindow Extended Attribute Synchronizer (DWEAS) is an interactive tool for updating many of the attributes of an existing DataWindow object with the current extended attributes from the PowerBuilder repository.

About DWEAS

The PowerBuilder repository contains extended attribute information for your database tables, including validation rules, display formats, edit styles, and text for column headings and labels.

When you create a DataWindow object, PowerBuilder uses information from the repository to create labels, headings, defaults, edit styles, display formats, and validation rules. But the relationship between an existing DataWindow and the repository is static: after a DataWindow is built, the DataWindow painter doesn't reference the repository. If an extended attribute changes in the repository, the DataWindow remains unchanged. The reason for this static design is that a dynamically linked repository can cause undesirable changes to DataWindow objects. DWEAS allows you to selectively update DataWindow objects with current extended attributes from the repository.

How DWEAS works

DWEAS uses PowerBuilder's dynamic DataWindow facility to create a new DataWindow object, which includes current extended attributes from the repository. Once created, DWEAS compares the new DataWindow object with the existing DataWindow object and highlights any differences. You can then selectively mark differing extended attributes and update the existing DataWindow object with the current repository values. Once the changes are applied, you can save the updated DataWindow object.

DWEAS examines the DataWindow's logical attributes only (column id, validation information, edit style information, and so on). The modification and placement of the physical attributes (text color, font, font size, and so on) depend on the standards in the specific development environments.

Using DWEAS

This section describes how to use DWEAS:

- Starting DWEAS
- Exiting DWEAS
- Logging on to your DBMS
- Loading a DataWindow
- The analysis process
- DWEAS workspace
- Identifying differences
- Saving changes
- Specifying DWEAS options
- DataWindow attributes modified by DWEAS
- DWEAS.INI file description

Starting DWEAS

❖ To start DWEAS, do one of the following:

- Double-click the DWEAS icon from the Windows Program Manager
- Double-click DWEAS.EXE in the Windows File Manager.
- Select File➤Run from the Windows Program Manager menu bar and type:

  ***directory name*\dweas**

The DWEAS About window displays. It closes after a five-second delay (or when you click OK or press ENTER).

Exiting DWEAS

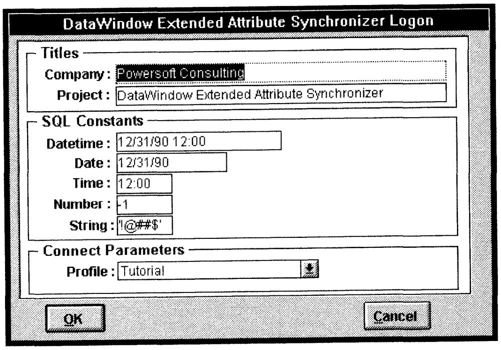
❖ To exit DWEAS, do one of the following:

- Press ALT+F4.
- Select File➤Exit from the menu bar.
- Click the Exit button on the DWEAS toolbar.
- Double-click the System menu in the top-left corner of the window.

If you have applied changes to a DataWindow and have not saved them, DWEAS will prompt you to save the changes.

Logging on to your DBMS

When you first start DWEAS, the DWEAS Logon dialog box displays.



DataWindow Extended Attribute Synchronizer Logon

Titles

Company : Powersoft Consulting

Project : DataWindow Extended Attribute Synchronizer

SQL Constants

Datetime : 12/31/90 12:00

Date : 12/31/90

Time : 12:00

Number : -1

String : '!@##$'

Connect Parameters

Profile : Tutorial

DWEAS uses the DWEAS.INI file to determine the initial values for the fields on this dialog box. When you enter new information in these fields and click OK, DWEAS.INI is updated (providing the logon is successful).

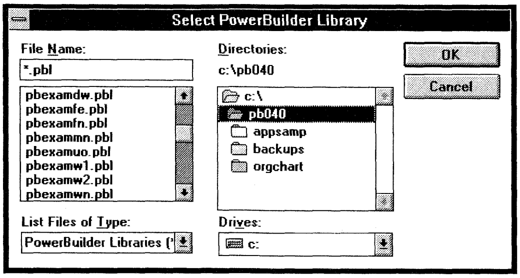
❖ To log on to your DBMS:

1. Verify that the Company and Project names are correct.
2. Verify that the profile name is correct; if necessary, change it using the Profile dropdown listbox.
3. Click OK.

Loading a DataWindow

After logging in to your DBMS, DWEAS prompts you for the library and DataWindow object to load. This is a two-step process: selecting a PBL file, and selecting an existing DataWindow object.

DWEAS displays the Select PowerBuilder Library dialog box and prompts you for a library selection.



File Name: *.pbl

pbexamdw.pbl
pbexamfe.pbl
pbexamfn.pbl
pbexammn.pbl
pbexamuo.pbl
pbexamw1.pbl
pbexamw2.pbl
pbexamwn.pbl

Directories: c:\pb040

c:\
pb040
appsamp
backups
orgchart

List Files of Type: PowerBuilder Libraries (*

Drives: c:

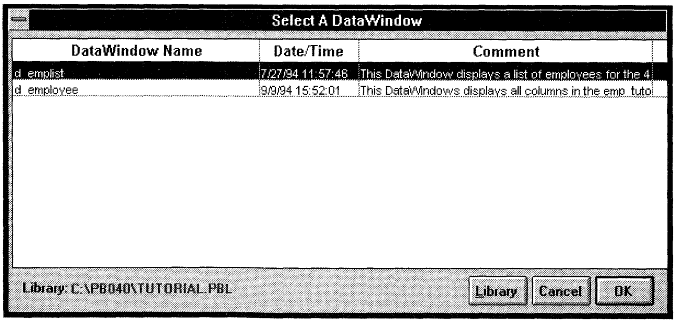
❖ To select a PBL file:

1 Use the Drives and Directories boxes to select the directory containing the desired PBL file.

2 Click on the desired PBL file.

3 Click OK.

After selecting a PBL file, DWEAS displays the Select A DataWindow dialog box.



DataWindow Name	Date/Time	Comment
d_emplist	7/27/94 11:57:46	This DataWindow displays a list of employees for the 4
d_employee	9/9/94 15:52:01	This DataWindows displays all columns in the emp_tuto

Library: C:\PB040\TUTORIAL.PBL

❖ To select a DataWindow object:

1 Click on the desired DataWindow object.

2. Click OK.

   DWEAS closes the dialog box and begins the analysis process.

To display DataWindow objects for another library
DWEAS displays the name of the selected directory in the lower-left corner of the window. To select a different library, click the Library button.

The analysis process

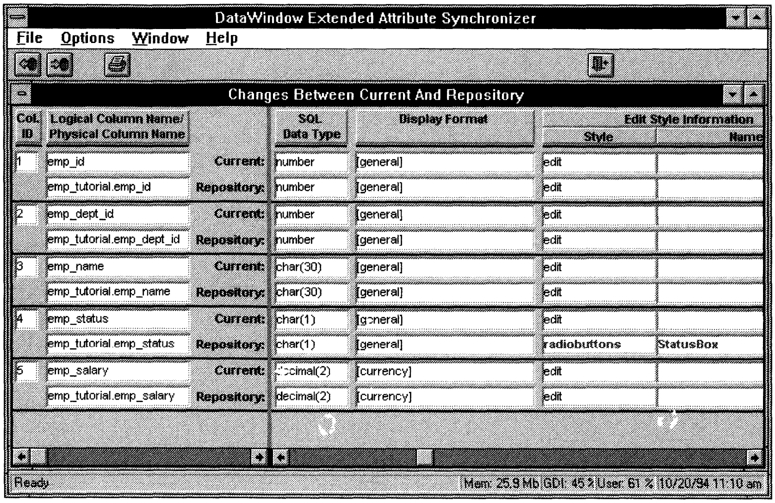
Once you select a DataWindow object, DWEAS loads and analyzes it. The time that it takes to perform the analysis depends on the speed of your machine, network, DBMS, and the number of columns in the DataWindow object.

DWEAS performs the analysis process by:

1. Loading the DataWindow object definition.
2. Creating an instance of the DataWindow object.
3. Accessing the SQL SELECT statement used to generate the DataWindow object.
4. Internally generating a Dynamic DataWindow object based on the SQL SELECT statement from step 3.
5. Analyzing the differences between each column of both DataWindow objects. This includes display formatting, validation rules, and edit style information.
6. Displaying the findings in the Differences Between Current And Repository window. Differences are highlighted with a special color, which you can modify.

DWEAS workspace

Once the analysis phase is complete, the DWEAS workspace displays. This is an example of the DWEAS workspace.


DataWindow Extended Attribute Synchronizer
File
Options
Window
Help
Changes Between Current And Repository

Col. ID	Logical Column Name/ Physical Column Name		SQL Data Type	Display Format	Edit Style Information	
					Style	Name
1	emp_id	Current:	number	[general]	edit	
	emp_tutorial.emp_id	Repository:	number	[general]	edit	
2	emp_dept_id	Current:	number	[general]	edit	
	emp_tutorial.emp_dept_id	Repository:	number	[general]	edit	
3	emp_name	Current:	char(30)	[general]	edit	
	emp_tutorial.emp_name	Repository:	char(30)	[general]	edit	
4	emp_status	Current:	char(1)	[g:neral]	edit	
	emp_tutorial.emp_status	Repository:	char(1)	[general]	radiobuttons	StatusBox
5	emp_salary	Current:	.':cimal(2)	[currency]	edit	
	emp_tutorial.emp_salary	Repository:	decimal(2)	[currency]	edit	

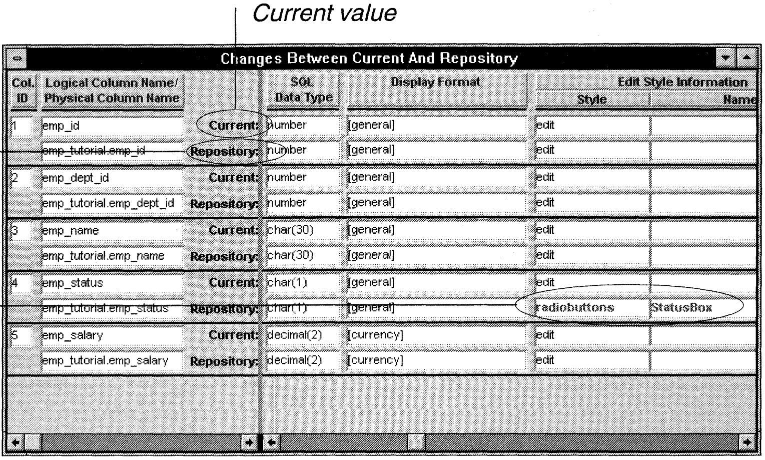
Ready Mem: 25.9 Mb GDI: 45 % User: 61 % 10/20/94 11:10 am

Identifying differences

This section describes how to identify differences between the current and repository attribute values and how to select attributes.

Changes Between Current and Repository window

You use the Changes Between Current and Repository window to view the results in order to identify attributes that need to be updated. This window displays the column identification information as well as the current and repository attribute values. The *current* values are the attributes as they exist in the selected DataWindow object, and the *repository* values are the attributes as they exist in the repository.



Current value

Repository value

Differing values are highlighted

Changes Between Current And Repository

Col. ID	Logical Column Name/ Physical Column Name		SQL Data Type	Display Format	Edit Style Information	
					Style	Name
1	emp_id	Current:	number	[general]	edit	
	emp_tutorial.emp_id	Repository:	number	[general]	edit	
2	emp_dept_id	Current:	number	[general]	edit	
	emp_tutorial.emp_dept_id	Repository:	number	[general]	edit	
3	emp_name	Current:	char(30)	[general]	edit	
	emp_tutorial.emp_name	Repository:	char(30)	[general]	edit	
4	emp_status	Current:	char(1)	[general]	edit	
	emp_tutorial.emp_status	Repository:	char(1)	[general]	radiobuttons	StatusBox
5	emp_salary	Current:	decimal(2)	[currency]	edit	
	emp_tutorial.emp_salary	Repository:	decimal(2)	[currency]	edit	

Selecting attributes

If there is a difference between the current and repository attribute values, DWEAS highlights the repository value and displays it in a different color, as specified in the Change Highlight Colors dialog box. When you update a DataWindow object with the repository value, DWEAS displays both values in the color associated with attributes that are marked for update.

❖ To update a DataWindow attribute with the repository value:

1 Double-click the repository value.

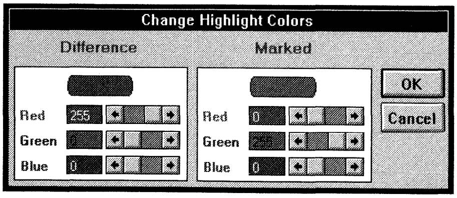
DWEAS displays the current and repository values in the color associated with updated attributes.

2 Continue updating DataWindow attributes until you are finished and ready to save the changes.

❖ To change the colors used to highlight differing attributes and attributes marked for update:

1 Select Options➤Change Colors from the menu bar.

The Change Highlight Colors dialog box displays.



Change Highlight Colors

Difference

Red 255
Green 0
Blue 0

Marked

Red 0
Green 255
Blue 0

2 Use the scrollbars to specify colors, as desired.

3 Click OK.

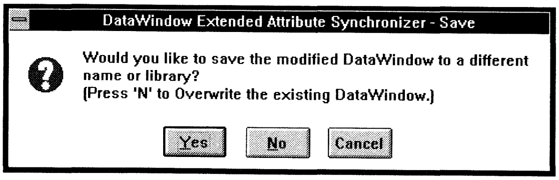
Saving changes

❖ To save the updated DataWindow:

1 Do one of the following:

- ♦ Click the Save button on the toolbar.
- ♦ Press CTRL+S.
- ♦ Select File➤Save from the menu bar.

DWEAS displays a dialog box asking if you want to overwrite the existing DataWindow object or save it with a new name:



DataWindow Extended Attribute Synchronizer - Save

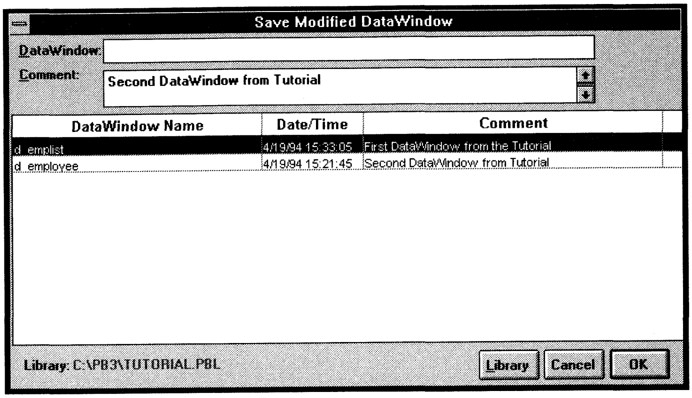
Would you like to save the modified DataWindow to a different name or library?
(Press 'N' to Overwrite the existing DataWindow.)

2 Click the option you want.

- ♦ **Yes** displays the Save Modified DataWindow dialog box.

  You use this dialog box to give a name to the updated DataWindow.

- **No** overwrites the existing DataWindow then reloads and reanalyzes the saved DataWindow object.
- **Cancel** redisplays the Changes between Current and Repository window.



Comment: Second DataWindow from Tutorial

DataWindow Name	Date/Time	Comment
d_emplist	4/19/94 15:33:05	First DataWindow from the Tutorial
d_employee	4/19/94 15:21:45	Second DataWindow from Tutorial

Library: C:\PB3\TUTORIAL.PBL

3 If you clicked Yes for step 2, the Save Modified DataWindow dialog box displays. Click the option you want.

- **OK** writes the DataWindow object to the selected library. It then reloads and reanalyzes the saved DataWindow object.
- **Library** displays the Select Destination PowerBuilder Library dialog box, which allows you to specify another library to contain the DataWindow object.
- **Cancel** redisplays the Changes between Current and Repository window.

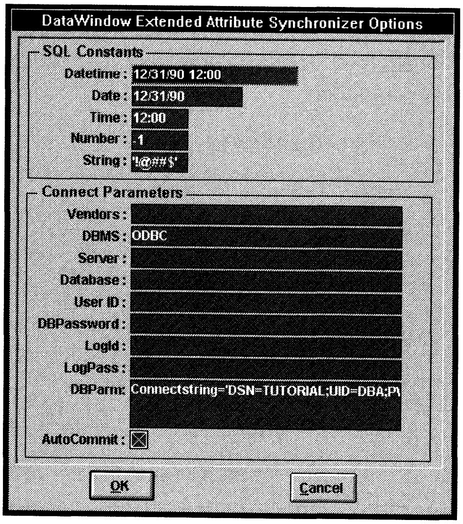
Specifying DWEAS options

DWEAS allows you to specify the constant values for retrieval argument replacement and the connection parameters for your DBMS.

❖ To specify DWEAS options:

1 Select Options➤System Options from the menu bar.

The DataWindow Extended Attribute Synchronizer Options dialog box displays.



DataWindow Extended Attribute Synchronizer Options

SQL Constants

Datetime : 12/31/90 12:00

Date : 12/31/90

Time : 12:00

Number : -1

String : '!@##$'

Connect Parameters

Vendors :

DBMS : ODBC

Server :

Database :

User ID :

DBPassword :

LogId :

LogPass :

DBParm: Connectstring='DSN=TUTORIAL;UID=DBA;P\

AutoCommit : ☒

2. Update options as desired:

   - **Use the SQL Constants section** to specify the values that DWEAS uses to replace retrieval arguments in the DataWindow SQL statements. The constants, which are used as dummy variables in the SQL statements, must be formatted properly for the target DBMS. Depending on the target DBMS, all constants except for Number may need to be enclosed in quotation marks.

   - **Use the Connect Parameters section** to provide the database connection information required by a PowerBuilder transaction object.

3. Click OK.

DataWindow attributes modified by DWEAS

DWEAS can modify all attributes specific to a particular edit style, including the name (for example, DWEAS can change *column*.Checkbox attributes to *column*.dddw attributes). Other attributes that DWEAS can modify include:

- *column*.BitmapName
- *column*.ColType
- *column*.Edit.Case
- *column*.Format
- *column*.Initial
- *column*.Validation
- *column*.ValidationMsg

For more information on DataWindow attributes, see the PowerBuilder *Function Reference.*

DWEAS.INI file description

The DWEAS.INI file controls various operational aspects of the DWEAS program. You update this file using the DataWindow Extended Attribute Synchronizer Options dialog box. There are four sections in DWEAS.INI: [Database], [Options], [Constants], and [Titles].

[Database] section

DWEAS uses the [Database] section to connect to the DBMS of your choice. This is a copy of the PB.INI file [Database] section.

Keyword	Meaning
Profile	Name of the Database you will use.
Vendors	List of DBMS vendors to which you will connect from DWEAS.
DBMS	A specific DBMS identifier from the Vendors list.
Server	If connecting to a remote DBMS, this is the server name or connect string. This value is dependent on the value entered in DBMS above.
Database	Name of the database to connect to (if required by the DBMS).
UserID	Identifier used to connect to the database.

Keyword	Meaning
Password	Default password used to connect to the database. (You can leave this blank or remove it for security purposes.)
AutoCommit	Depending on your DBMS, you may need to change this from its default state of FALSE (0) to TRUE (1).
DBParm	DBMS-dependent keyword typically used for ODBC.

[Options] section

DWEAS uses the [Options] section to hold variables that affect its presentation styles.

Variable	Result
ChangeColor	RGB color value used to denote an attribute whose value is different in the repository when compared to the current DataWindow object
MarkedColor	RGB color value used to denote an attribute that has been selected to be modified in the current DataWindow object

[Constants] section

DWEAS uses the [Constants] section when it is constructing sample data for populating the Current DataWindow object. You may need to change this, depending on your DBMS.

Variable	Result
date	Date value in a format consistent with the selected DBMS
time	Time value in a format consistent with the selected DBMS
datetime	DateTime value in a format consistent with the selected DBMS
number	Numeric value in a format consistent with the selected DBMS
string	Character (Char/Varchar) value in a format consistent with the selected DBMS

[Titles] section

DWEAS uses the [Titles] section to keep the information used in the header of the DWEAS Report.

Keyword	Meaning
Company	String containing the company name displayed in the DWEAS Report header
Project	String containing the project name displayed in the DWEAS Report header

DWEAS menu bar

DWEAS contains a menu bar and a toolbar. This section describes the menu bar and corresponding toolbar items unique to DWEAS.

☞ For information on the Window and Help menus, see the discussions in Chapter 2, "Cross Reference."

File menu

The File menu contains menu choices used to retrieve, save, and print the DataWindow objects.

Menu item	Activity	Shortcut key	Toolbar button
Load	Loads an existing DataWindow from a PowerBuilder Library	CTRL+L	
Save	Applies updated attributes to the DataWindow object and saves the DataWindow to a PowerBuilder Library	CTRL+S	
Print	Prints the active window	CTRL+P	
Print Preview	Displays the active window in preview mode		
Print Setup	Displays the Windows Printer Setup dialog box	CTRL+T	
Exit	Exits DWEAS	ALT+F4	

Options menu

The Options menu contains choices that allow you to make changes to DWEAS colors and to modify system settings.

Menu item	Activity
Change Colors	Displays the Change Highlight Colors dialog box, which allows you to modify the colors that highlight differing attribute values in the Changes Between Current and Repository window
System Options	Displays the DataWindow Extended Attribute Synchronizer Options dialog box, which allows you to specify constant values for retrieval argument replacements and the connection parameters for your DBMS

CHAPTER 4

DataWindow SQL Verifier

About this chapter This chapter describes the DataWindow SQL Verifier.

Contents

DataWindow SQL Verifier overview

The DataWindow SQL Verifier checks the validity of SQL statements that are used by the DataWindow objects in a PowerBuilder application.

About DataWindow SQL Verifier

When an application designed for one DBMS is migrated to another DBMS, it may be necessary to check the SQL statements originally created when the DataWindows were built. For example, one DBMS may require that all SQL statements be enclosed in quotation marks; another may flag quotation marks as invalid. Without the DataWindow SQL Verifier, you would have to run the application and test each DataWindow against the different target DBMSs to determine if any problems existed.

You can also use the DataWindow SQL Verifier to determine if columns have been added or deleted from database tables.

Once the DataWindow SQL Verifier analyzes the DataWindow objects in an application, you can view and print the results.

How the DataWindow SQL Verifier works

The DataWindow SQL Verifier works at an application level. First, you specify what application to examine. Then the DataWindow SQL Verifier logs on to the DBMS, extracts the SQL statement from each of the application's DataWindow objects, executes it against the selected DBMS, and generates a report identifying any problems.

Typical errors that the DataWindow SQL Verifier highlights are invalid column names and invalid table names. The DataWindow SQL Verifier also ensures that updatable columns in the DataWindow object match column names in the target DBMS. For a more detailed analysis of the DataWindow at a structural level, use the DataWindow Extended Attribute Synchronizer (DWEAS).

Using the DataWindow SQL Verifier

This section describes how to use the DataWindow SQL Verifier:

- Starting the DataWindow SQL Verifier
- Exiting the DataWindow SQL Verifier
- DataWindow SQL Verifier workspace
- Selecting an application
- The verification process
- Viewing the results
- Specifying DataWindow SQL Verifier options
- DWCHECK.INI file description

Starting the DataWindow SQL Verifier

❖ To start the DataWindow SQL Verifier, do one of the following:

- Double-click the DataWindow SQL Verifier icon from the Windows Program Manager.
- Double-click DWCHECK.EXE in the Windows File Manager.
- Select File➢Run from the Windows Program Manager menu bar and type:

  ***directory name*\dwcheck**

The DataWindow SQL Verifier banner displays until initialization completes.

Exiting the DataWindow SQL Verifier

❖ To exit the DataWindow SQL Verifier, do one of the following:

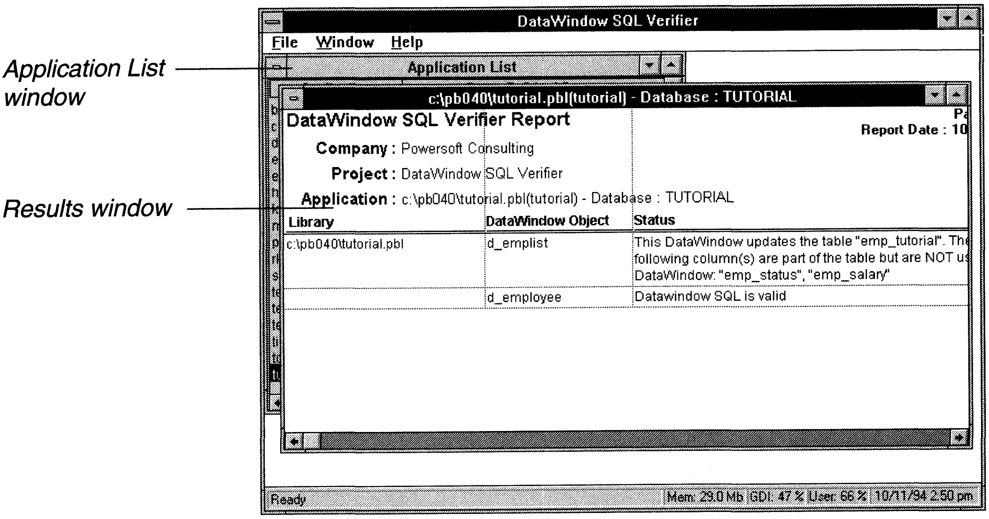
- Press ALT+F4.
- Select File➢Exit from the menu bar.

- Double-click the System menu in the top-left corner of the window.

Print reports before exiting
If you have generated DataWindow SQL Verifier reports that you want to keep, be sure to print them before exiting. When you exit the DataWindow SQL Verifier, all results are deleted.

DataWindow SQL Verifier workspace

Here is an example of the DataWindow SQL Verifier workspace.


Application List window
Results window
DataWindow SQL Verifier
File Window Help
Application List
c:\pb040\tutorial.pbl(tutorial) - Database : TUTORIAL
Ready
Mem: 29.0 Mb GDI: 47 % User: 66 % 10/11/94 2:50 pm

DataWindow SQL Verifier Report

Pa

Report Date : 10

Company : Powersoft Consulting

Project : DataWindow SQL Verifier

Application : c:\pb040\tutorial.pbl(tutorial) - Database : TUTORIAL

Library	DataWindow Object	Status
c:\pb040\tutorial.pbl	d_emplist	This DataWindow updates the table "emp_tutorial". The following column(s) are part of the table but are NOT us DataWindow: "emp_status", "emp_salary"
	d_employee	Datawindow SQL is valid

The Application List window lists the application objects and the PBL files in which they reside. The DataWindow SQL Verifier displays a separate results window for each selected application.

If the Application List window is empty
The DataWindow SQL Verifier uses the PB.INI file to obtain the application list. If the PB.INI file cannot be found, a message displays with an empty Application List window. This problem may indicate that the Advanced PowerBuilder Utilities were not installed in the PowerBuilder directory.

If a "Cannot read from drive" message displays
The DataWindow SQL Verifier checks to see if each application named in the PB.INI file exists. If this application resides on a drive that is unavailable (such as an empty diskette drive or an unattached network drive), file errors will interrupt the process. Click Cancel to continue.

Selecting an application

Once the DataWindow SQL Verifier is running, you start the verification process by selecting a PowerBuilder application from the Application List window.

❖ To verify DataWindow SQL for an application:

- Select an application from the Application List window by doing one of the following:
  - Double-click on the desired application.
  - Highlight an application row using the keyboard UP or DOWN ARROW keys and then press ENTER.

During the verification process, a progress window displays the percentage of the DataWindows that have been examined. This progress window also contains a Cancel button, which allows you to cancel the verification process.

The verification process

Once you select an application object, the DataWindow SQL Verifier logs on to the DBMS, displays an empty results window, and begins analysis of each DataWindow object in each library that is used in the application. The time it takes to perform the analysis depends on the speed of your machine and network, and the number of DataWindow objects in the selected application.

The DataWindow SQL Verifier process scans the application's libraries for DataWindow objects, displaying each DataWindow object name on the status line and in the results window. It then performs the following steps for each DataWindow object:

1 The SQL statement is retrieved and examined. If any retrieval arguments were specified, they are replaced with constant values from DWCHECK.INI. You can change these constants using the DataWindow SQL Verifier Options dialog box; for how, see page 46.

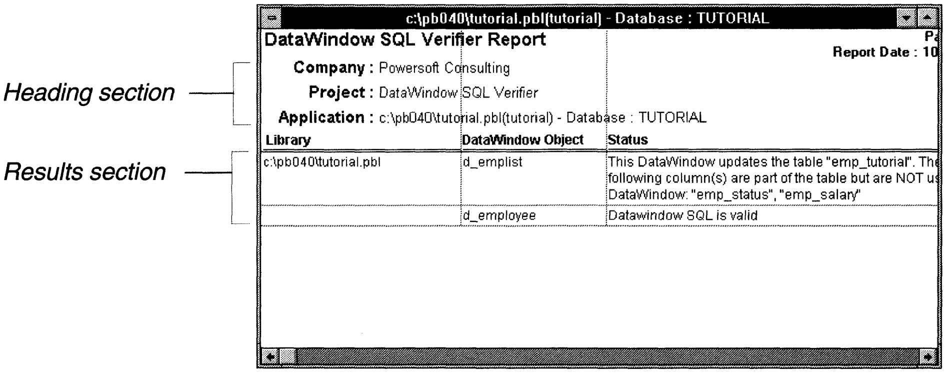
2 The SQL statement is submitted to the DBMS and the result is checked for errors.

3 The DataWindow SQL Verifier displays the status in the results window.

4 This process continues until all DataWindow objects in the selected application have been tested.

Viewing the results

You can view the results of the DataWindow SQL Verifier in the results window.



c:\pb040\tutorial.pbl(tutorial) - Database : TUTORIAL

DataWindow SQL Verifier Report

P
Report Date : 10

Company : Powersoft Consulting
Project : DataWindow SQL Verifier
Application : c:\pb040\tutorial.pbl(tutorial) - Database : TUTORIAL

Library	DataWindow Object	Status
c:\pb040\tutorial.pbl	d_emplist	This DataWindow updates the table "emp_tutorial". Th following column(s) are part of the table but are NOT u DataWindow: "emp_status", "emp_salary"
	d_employee	Datawindow SQL is valid

The title of the results window is determined by the main PBL file and name of the selected application. The DataWindow SQL Verifier results window contains a heading section that describes the company, name of the project, and name of the application (including its library).

The detail section of the results window contains rows for each DataWindow object that was analyzed. The rows are divided into three columns:

Column	Meaning
Library	The name of the PowerBuilder Library file (PBL) that contains the referenced DataWindow object. This column is blank when the library is the same as the preceding one.
DataWindow Object	The name of the referenced DataWindow object.
Status	The result of the check of the DataWindow SQL. One of the following messages displays: **DataWindow SQL Is Valid** – The SQL was executed without producing any errors. **Script Source DataWindow** – The DataWindow is script-driven and contains no SQL statement. **Missing column** – Column named in the DataWindow no longer exists in the database. **Unable to create DataWindow** – DataWindow could not be generated: possible version mismatch. ***Incomplete update message*** – Your DataWindow updates the database but does not access all database columns. This may not be a problem, however, it could indicate that columns have been added to the table since the DataWindow was created. ***DBMS-specific error message*** – An error occurred while executing the SQL statement; the message returned by the DBMS is displayed here.

At this point, you may view or print the results of the DataWindow SQL Verifier, modify headings, or select another application to analyze.

❖ To change the Company and/or Project Name in the report heading:

1 Select File➢Options from the menu bar.

The DataWindow SQL Verifier Options dialog box displays.

2 Make your changes.

3 Click OK.

The changes are immediately reflected in the current report.

❖ To analyze another application:

1 Display the Application List window.

2 Select an application from the Application List window in one of two ways:

- ♦ Double-click on the selected application.
- ♦ Highlight an application row using the keyboard UP or DOWN ARROW keys and then press ENTER.

The DataWindow SQL Verifier connects to the DBMS and analyzes DataWindow objects in the selected application.

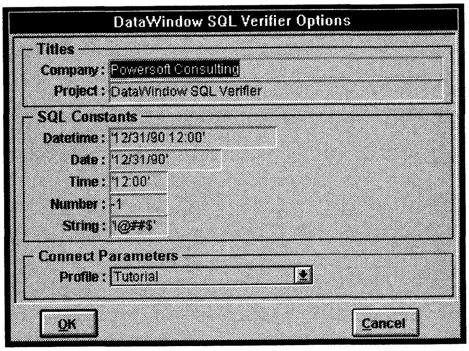
Specifying DataWindow SQL Verifier options

You can adjust default settings using the DataWindow SQL Verifier Options dialog box.

❖ To control DataWindow SQL Verifier options:

1 Select File➤Options from the menu bar.

The DataWindow SQL Verifier Options dialog box displays.



Company : Powersoft Consulting

Project : DataWindow SQL Verifier

Datetime : '12/31/90 12:00'

Date : '12/31/90'

Time : '12:00'

Number : -1

String : '!@#$'

Profile : Tutorial

2 Specify options, as necessary:

- **The Titles section** allows you to specify the Company Name and Project Name that display in the report heading.
- **The SQL Constants** section allows you to specify the values that replace retrieval arguments in the DataWindow SQL Statements. All constants, except for Number, must be enclosed in quotation marks. Also, the constants must be formatted correctly for the target DBMS. To speed the verification process, you should ensure that these constants do not return a result set.
- **The Connect Parameters** section allows you to select a database connection profile as defined in the PB.INI file.

3 Click OK.

DWCHECK.INI file description

The DWCHECK.INI file controls various operational aspects of the DataWindow SQL Verifier program. You update this file using the DataWindow SQL Verifier Options dialog box. There are three sections in DWCHECK.INI: [Database1], [Constants], and [Titles].

[Database1] section

The [Database1] section contains information on the current DBMS connection. You specify the Profile keyword using the DataWindow SQL Verifier Options dialog box. The other keywords are copied automatically from the PB.INI file.

Keyword	Meaning
Profile	Name of the profile used in the PB.INI file.
DBMS	Specific DBMS identifier from the list after the Vendors keyword in the PB.INI file.
ServerName	Server name or connect string, if connecting to a remote DBMS. This value is dependent on the value entered in DBMS above.
Database	Name of the database to connect to (for DBMSs that require it).
UserID	Identifier used to connect to the database.
Databasepassword	Password that is associated with the UserID.
LogID	Identifier used to connect to the database.
Logpass	Password that is associated with the LogID.

Keyword	Meaning
DBParm	DBMS-dependent keyword; typically used for ODBC.

[Constants] section

The [Constants] section contains values that the DataWindow SQL Verifier uses to replace retrieval arguments in the DataWindow SQL statements. The keywords in this section are DBMS-dependent; you may need to change them depending on the selected DBMS. To speed the verification process, you should ensure that these constants do not return a result set.

Keyword	Result	Format
date	Date value in a format consistent with your selected DBMS	Enclose in quotation marks
time	Time value in a format consistent with your selected DBMS	Enclose in quotation marks
datetime	DateTime value in a format consistent with your selected DBMS	Enclose in quotation marks
number	Numeric value in a format consistent with your selected DBMS	No quotation marks
string	Character (Char/Varchar) value in a format consistent with your selected DBMS	Enclose in quotation marks

[Titles] section

The [Titles] section contains the information in the header of the DataWindow SQL Verifier Report.

Keyword	Meaning
Company	String containing the name of the company displayed in the DataWindow SQL Verifier Report header
Project	String containing the name of the project displayed in the DataWindow SQL Verifier Report header

DataWindow SQL Verifier menu bar

The DataWindow SQL Verifier contains a menu bar and a toolbar. This section describes the menu items unique to the DataWindow SQL Verifier.

☞ For information on the Window and Help menus, see the discussions in Chapter 2, "Cross Reference."

File menu

The File menu performs actions on the results of the DataWindow SQL Verifier and specifies execution options.

Menu item	Activity	Shortcut key
Options	Displays the DataWindow SQL Verifier Options dialog box, which allows you to set options for the DataWindow SQL Verifier	
Print	Prints the current DataWindow SQL Verifier Report or Application List window	
Print Setup	Displays the standard Windows Printer Setup dialog box	
Close	Closes the current window. You cannot close the Application List window	CTRL+F4
Exit	Exits the DataWindow SQL Verifier	ALT+F4

CHAPTER 5

PowerBuilder Extended Attribute Reporter (PEAR)

About this chapter

This chapter describes the PowerBuilder Extended Attribute Reporter (PEAR).

Contents

PEAR overview

The PowerBuilder Extended Attribute Reporter (PEAR) reports on extended attributes contained in the PowerBuilder repository.

About PEAR

PEAR displays extended attributes for one or more chosen database tables, including labels, headers, justification, case, display format, validation rules, edit styles, and initial value.

How PEAR works

PEAR accesses your database's system tables to determine tables that can be displayed. It also accesses the system tables to determine the columns in tables selected for reporting. PEAR then accesses the PowerBuilder repository and displays extended attributes for all columns in the selected tables.

PEAR.INI contains system table information
The PEAR.INI file contains DBMS-specific syntax that PEAR uses to access the system tables that contain database table and column definitions.

☞ For more information, see "PEAR.INI file description" on page 60.

Using PEAR

This section describes how to use PEAR:

- Starting PEAR
- Exiting PEAR
- Connecting to your DBMS
- PEAR workspace
- Selecting tables
- Viewing the results
- Specifying PEAR options
- PEAR.INI file description

Starting PEAR

❖To start PEAR do one of the following:

- Double-click the PEAR icon from the Windows Program Manager.
- Double-click PEAR.EXE in the Windows File Manager.
- Select File➢Run from the menu bar of the Windows Program Manager and type the following:

  ***directory name*\pear**

The PowerBuilder Extended Attribute Reporter banner displays until PEAR finishes initializing.

Exiting PEAR

❖ To exit PEAR do one of the following:

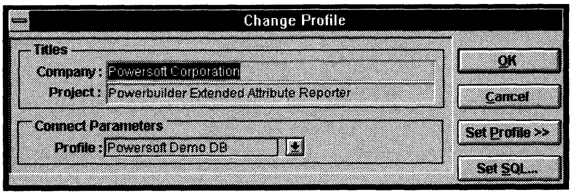
- Press ALT+F4.
- Select File➢Exit from the menu bar.
- Click the Exit button on the toolbar.

- Double-click the system menu in the top-left corner of the window.

> **Print reports before exiting**
> If you have generated PEAR reports that you want to keep, be sure to print them before exiting. When you exit PEAR, all results are deleted.

Connecting to your DBMS

When you first start PEAR, the Change Profile dialog box displays.



Change Profile

Titles

Company: Powersoft Corporation

Project: Powerbuilder Extended Attribute Reporter

Connect Parameters

Profile: Powersoft Demo DB

PEAR uses the PEAR.INI file to determine the initial values for the fields in this dialog box. When you enter new information in these fields and click OK, PEAR.INI is updated (provided the logon is successful).

☞ For more information on the Change Profile dialog box, see "Specifying PEAR options" on page 58.

❖ To connect to a database:

1 Verify that the Company and Project names are correct.

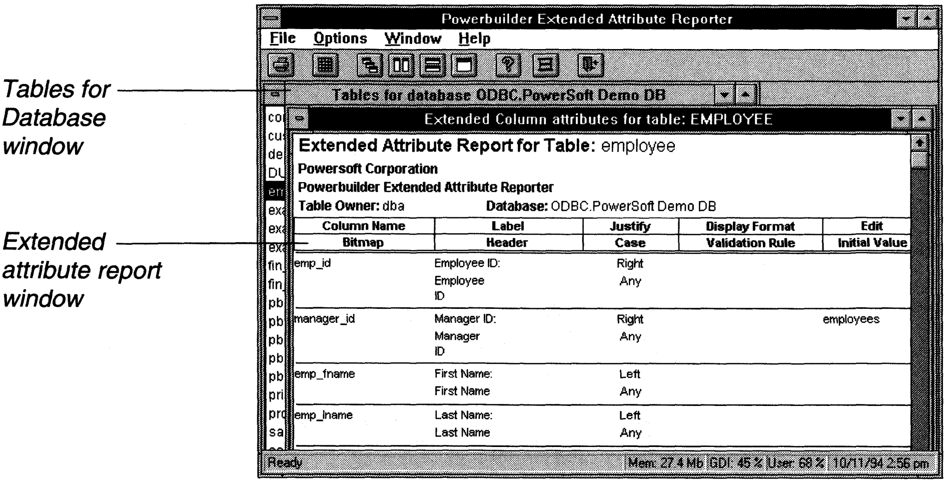
2 Verify that the profile name is correct; if necessary, change it using the Profile dropdown listbox.

3 Click OK.

PEAR connects to the specified database and displays the Tables for Database window.

PEAR workspace

The PEAR main window is shown below.


Tables for Database window
Extended attribute report window
Powerbuilder Extended Attribute Reporter
File Options Window Help
Tables for database ODBC.PowerSoft Demo DB
co
cu
de
DU
em
ex
ex
ex
fin
fin
pb
pb
pb
pb
pb
pri
pr
sa
Extended Column attributes for table: EMPLOYEE
Ready
Mem: 27.4 Mb GDI: 45 % User: 68 % 10/11/94 2:56 pm

Extended Attribute Report for Table: employee

Powersoft Corporation

Powerbuilder Extended Attribute Reporter

Table Owner: dba **Database:** ODBC.PowerSoft Demo DB

Column Name	Label	Justify	Display Format	Edit
Bitmap	**Header**	**Case**	**Validation Rule**	**Initial Value**
emp_id	Employee ID:	Right		
	Employee ID	Any		
manager_id	Manager ID:	Right		employees
	Manager ID	Any		
emp_fname	First Name:	Left		
	First Name	Any		
emp_lname	Last Name:	Left		
	Last Name	Any		

The Tables for Database window lists the tables named in the DBMS system tables, as specified in the PEAR.INI file. PEAR displays a separate extended attribute report window for each selected table.

Selecting tables

Once PEAR is running, you use the Tables for Database window to select one or more tables for reporting.

❖ To select tables from the Tables for Database window do one of the following:

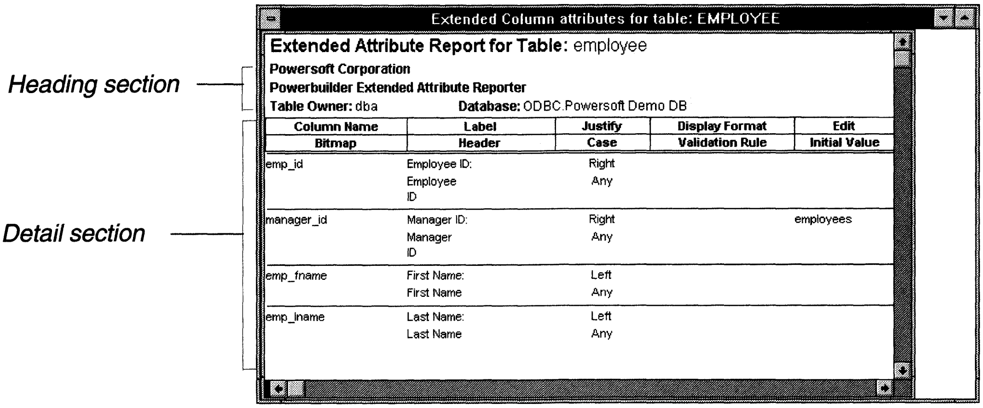
- Double-click on the table name.
- Highlight an application row using the UP ARROW or DOWN ARROW keys and then press ENTER.
- Select multiple rows of tables (using SHIFT+click for two or more tables in sequence and CTRL+click for two or more tables out of sequence), then select File➤Run Report from the menu bar. This opens a results window for each selected table.

When there is an error retrieving information
This window displays the table names from the respective system tables of the current database. If there is an error retrieving information, check the PEAR.INI file for the proper SQL syntax used to modify this window.

For more information, see "PEAR.INI file description" on page 60.

Viewing the results

Once you select a table, PEAR accesses the repository and displays the Extended Column Attributes window, which shows extended attribute information for each column.



Heading section

Detail section

Extended Column attributes for table: EMPLOYEE

Extended Attribute Report for Table: employee

Powersoft Corporation
Powerbuilder Extended Attribute Reporter
Table Owner: dba **Database:** ODBC.Powersoft Demo DB

Column Name	Label	Justify	Display Format	Edit
Bitmap	**Header**	**Case**	**Validation Rule**	**Initial Value**
emp_id	Employee ID:	Right		
	Employee ID	Any		
manager_id	Manager ID:	Right		employees
	Manager ID	Any		
emp_fname	First Name:	Left		
	First Name	Any		
emp_lname	Last Name:	Left		
	Last Name	Any		

Use the scrollbars to see the entire report.

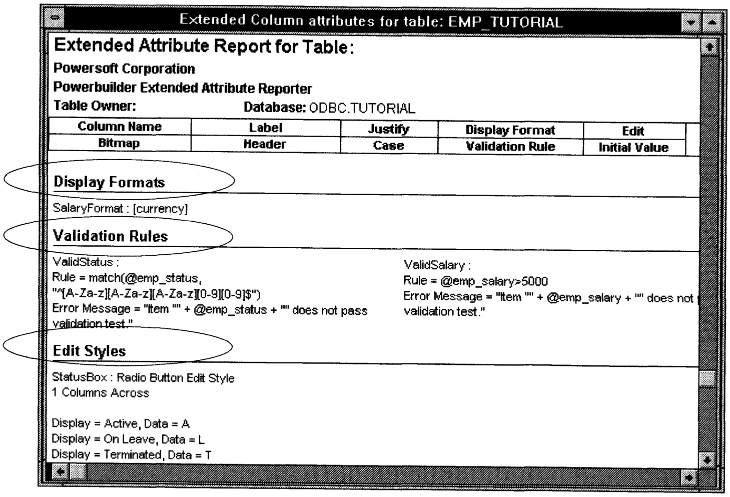
The title of the results window is determined by the selected table. The PEAR results window contains a heading section that describes the company, name of the project, table owner, and database.

The detail section of the results window contains two rows for each column in the table that was analyzed. The rows are divided into these columns:

Column	Meaning
Column name	Name of the column
Bitmap	Bitmap, if defined for the column
Label	Label for use with freeform DataWindows

Column	Meaning
Header	Heading for use with tabular DataWindows
Justify	Justification (left, right, center)
Case	Case (upper, lower, any)
Display Format	Name of display format, if defined for the column
Validation Rule	Name of validation rule, if defined for the column
Edit	Edit style, if defined for the column
Initial Value	Initial value, if defined for the column

The last section of the report provides a summary of display formats, validation rules, and edit styles.



Extended Column attributes for table: EMP_TUTORIAL

Extended Attribute Report for Table:

Powersoft Corporation

Powerbuilder Extended Attribute Reporter

Table Owner: **Database:** ODBC.TUTORIAL

Column Name	Label	Justify	Display Format	Edit
Bitmap	Header	Case	Validation Rule	Initial Value

Display Formats

SalaryFormat : [currency]

Validation Rules

```
ValidStatus :
Rule = match(@emp_status,
"^[A-Za-z][A-Za-z][A-Za-z][0-9][0-9]$")
Error Message = "Item "" + @emp_status + "" does not pass
validation test."
```

```
ValidSalary :
Rule = @emp_salary>5000
Error Message = "Item "" + @emp_salary + "" does not
validation test."
```

Edit Styles

```
StatusBox : Radio Button Edit Style
1 Columns Across

Display = Active, Data = A
Display = On Leave, Data = L
Display = Terminated, Data = T
```

❖ To change the Company Name and/or Project Name in the Report heading:

1 Select Options➤Configure Profile from the menu bar.

2 Make your changes.

3 Click OK.

The changes are immediately reflected in the current report.

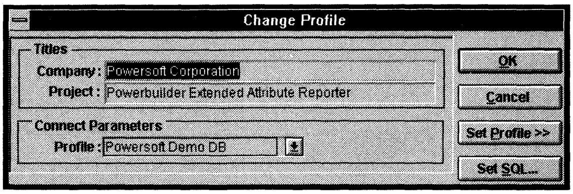
Specifying PEAR options

PEAR allows you to change heading, profile, and database parameter defaults.

❖ To specify PEAR options:

1 Select Options➤Change Profile from the menu bar.

The Change Profile dialog box displays showing the current connection information.



Titles
Company: Powersoft Corporation
Project: Powerbuilder Extended Attribute Reporter

Connect Parameters
Profile: Powersoft Demo DB

PEAR access profiles from the PB.INI file. If you have no PB.INI file, the current profile stored in PEAR.INI is displayed.

2 Change options as necessary:

- **The Titles section** allows you to specify the Company Name and Project Name that go into the results report heading.
- **The Connect Parameters** section allows you to select a database connection profile as defined in the PB.INI file.

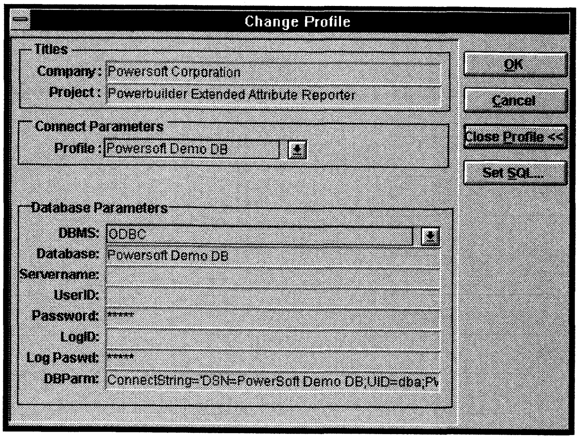
3 Click the desired button:

- **OK** updates the PEAR.INI file with the new specifications and closes the dialog box.
- **Cancel** closes the dialog box and returns to the window without applying changes.
- **Set Profile** displays an expanded Set Profile dialog box, which allows you to set up a database connection and choose a new profile.
- **Set SQL** displays Set SQL for Database dialog box, which allows you to modify the SQL= and COL= lines for the current DBMS (from the PEAR.INI file).

❖ To specify detailed database information:

1 Click the Set Profile button.

The expanded Set Profile dialog box displays. This dialog box allows you to specify detailed database information in the Database Parameters box.



Company: Powersoft Corporation
Project: Powerbuilder Extended Attribute Reporter

Profile: Powersoft Demo DB

DBMS: ODBC
Database: Powersoft Demo DB
Servername:
UserID:
Password: *****
LogID:
Log Paswd: *****
DBParm: ConnectString='DSN=PowerSoft Demo DB;UID=dba;P\

In the profile example above, the current profile is set to Tutorial, which will attach the user to an ODBC (Watcom) database called TUTORIAL. You can change the information in this window and select a new profile to attach to any available DBMS.

2 Update database information as needed.

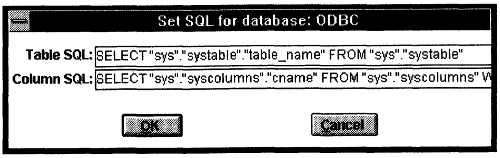
3 Click the desired button.

- **OK** updates the PEAR.INI file with the new specifications and closes the dialog box.
- **Cancel** closes the dialog box and returns to the window without applying changes.
- **Close Profile** displays the condensed Set Profile dialog box.
- **Set SQL** displays Set SQL for Database dialog box, which allows you to modify the SQL= and COL= lines for the current DBMS (from the PEAR.INI file).

❖ To modify SQL= and COL= specifications:

1 With the Change Profile dialog box displayed, click Set SQL.

The Set SQL for Database dialog box displays.



Set SQL for database: ODBC

Table SQL: SELECT "sys"."systable"."table_name" FROM "sys"."systable"

Column SQL: SELECT "sys"."syscolumns"."cname" FROM "sys"."syscolumns" W

2 Modify SQL statements, as needed.

3 Click OK.

PEAR.INI file description

The PEAR.INI file controls information about the current database and to store reporting information. You update this file using the Change Profile dialog box. There are multiple sections in PEAR.INI: [Company], [Database], [Pbtables], and one section for each DBMS data source type.

[Company] section

The [Company] section stores reporting information. The company name and project name are printed on all extended attribute reports. If no company or project name is supplied, the report headings will be blank.

Keyword	Meaning
Company	String that represents the company name that is displayed in the Extended Column Attributes report header
Project	String that represents the project name that is displayed in the Extended Column Attributes report header

[Database] section

PEAR uses the [Database] section to connect to the selected DBMS. It copies most of the keywords used in the [Database] section of the PB.INI file.

Keyword	Meaning
Profile	Name of the current profile.
DBMS	Specific DBMS identifier from the list after the vendors keyword.
Servername	Server name or connect string, if connecting to a remote DBMS. This value is dependent on the value entered in DBMS above.
Database	Name of the database to connect to (for DBMSs that require it).
UserID	Identifier used to connect to the database.
Databasepassword	Password used (if needed) to connect to the database.
LogID	Identifier used to connect to the database.
LogPassword	Log password used to connect to the database.
DBParm	DBMS-dependent keyword; typically used for ODBC.

[Pbtables] section

PEAR uses the [Pbtables] section to connect to the PowerBuilder system tables. If your DBMS requires, you may need to add owner name qualification to all tables.

Keyword	Meaning
Pbcatedt	SQL SELECT statement to access PowerBuilder edit styles
Pbcatfmt	SQL SELECT statement to access PowerBuilder display formats
Pbcatvld	SQL SELECT statement to access PowerBuilder validation rules
Pbcatcol	SQL SELECT statement to access PowerBuilder extended attribute information

DBMS-specific section

This section stores DBMS-specific SQL statements used by the Tables for Database and Extended Attribute windows to access table and column information from the DBMS system tables.

- **The Tables for Database window** uses the SQL= specification to create a SQL SELECT statement to display all tables in a database

- **The Extended Attribute window** uses the COL= specification to create a SQL SELECT statement to display the columns associated with the chosen table

For each DBMS's section, you specify two lines.

Keyword	Meaning
SQL	Specifies a SQL SELECT statement that returns a list of tables defined to the database. This statement must return one column: the table name.
COL	Specifies a SQL SELECT statement that returns a list of columns for a particular table. This statement must return one column: the column name. Additionally, the statement must include a retrieval argument for table name and the argument must be **:table_name**

PEAR.INI includes sections for many popular DBMSs. If the SQL= and COL= statements are not correct for your DBMS, change them, as necessary, to return table names and column names. You may need to add a section for your particular DBMS.

> **ORACLE users**
> ORACLE users must copy the lines from the ORACLE6 or ORACLE7 sections, as appropriate, and paste them under the ORACLE section.

You can specify additional WHERE criteria to further restrict the tables and columns displayed. For example, you might add restrictions to the SQL= specification to omit DBMS and PowerBuilder system tables.

This is an example of a DBMS-specific section.

```
[SYBASE]
sql=SELECT sysobjects.name FROM sysobjects WHERE
type = 'U' order by sysobjects.name
col=SELECT syscolumns.name FROM syscolumns WHERE
syscolumns.tbname = :table_name
```

PEAR menu bar

PEAR contains a menu bar and a toolbar. This section outlines the menu bar and corresponding toolbar items unique to PEAR.

☞ For information on the Window and Help menus, see the discussions in Chapter 2, "Cross Reference."

File menu

The File menu contains menu choices used to retrieve and print extended attribute information.

Menu item	Activity	Shortcut key	Toolbar button
Run Report	Runs PEAR, creating an extended attribute report	CTRL+R	
Print	Prints the current window	CTRL+P	🖨
Print Preview	Shows you what the report will look like when you print it		
Printer Setup	Displays the standard Windows Printer Setup dialog box		
Close	Closes the current window		
Exit	Exits PEAR	ALT+F4	🚪

Options menu

The Options menu contains choices that allow you to modify default settings.

Menu item	Activity	Toolbar button
Configure Profile	Displays the Change Profile dialog box, which allows you to specify report headings and change the current database profile	▦

CHAPTER 6

Watcom Image Editor

About this chapter

This chapter describes the Watcom Image Editor.

Contents

Image Editor overview

You can use the Watcom Image Editor to create and modify icons, bitmaps, and cursors.

About the Image Editor

The Image Editor is a visual drawing tool with the power to draw all the visual resources you want, exactly the way you picture them. You can use the Image Editor to design effective bitmaps, meaningful icons, and communicative cursors.

The Image Editor can also copy visual images from your computer screen. For example, if you would like to incorporate a favorite bitmap or icon from the Windows desktop or another application, you can use the Image Editor's Snap Bitmap option to copy that image to the design grid. Then you can tailor that image to your exact specifications. When you're happy with the visual image, you can save it as a bitmap, icon, or cursor.

Using visual images in PowerBuilder

You can use the visual images created using the Watcom Image Editor as PowerBuilder bitmaps, icons, and cursors.

- You can use bitmaps in windows as Picture and Picture Button controls.
- You can use bitmaps in menus as toolbar buttons.
- You can create icons to represent applications and minimized windows.
- You can specify different cursors (also called pointers) for different fields in Windows and DataWindows.

Exception
You cannot add these icons to PowerBuilder's PowerBar, PowerPanel, or PainterBar. You can, however, associate bitmaps with your MDI application's menu items to create visually pleasing toolbars in your own applications.

Using the Image Editor

This section describes how to use the Image Editor to create icons, bitmaps, and cursors.

- Starting the Image Editor
- Exiting the Image Editor
- The main window
- Creating a design grid
- The toolbox
- Creating and editing bitmaps
- Creating and editing cursors
- Creating and editing icons
- Copying images
- Selecting screen and inverse colors
- Assigning colors to mouse buttons
- Using the color palette
- Changing colors
- Customizing current settings

Starting the Image Editor

❖ To start the Image Editor, do one of the following:

- Double-click the Image Editor icon from the Windows Program Manager.
- Double-click WIMGEDIT.EXE in the Windows File Manager.
- Select File➤Run from the menu bar of the Windows Program Manager and type the following:

  ***directory name*\wimgedit**

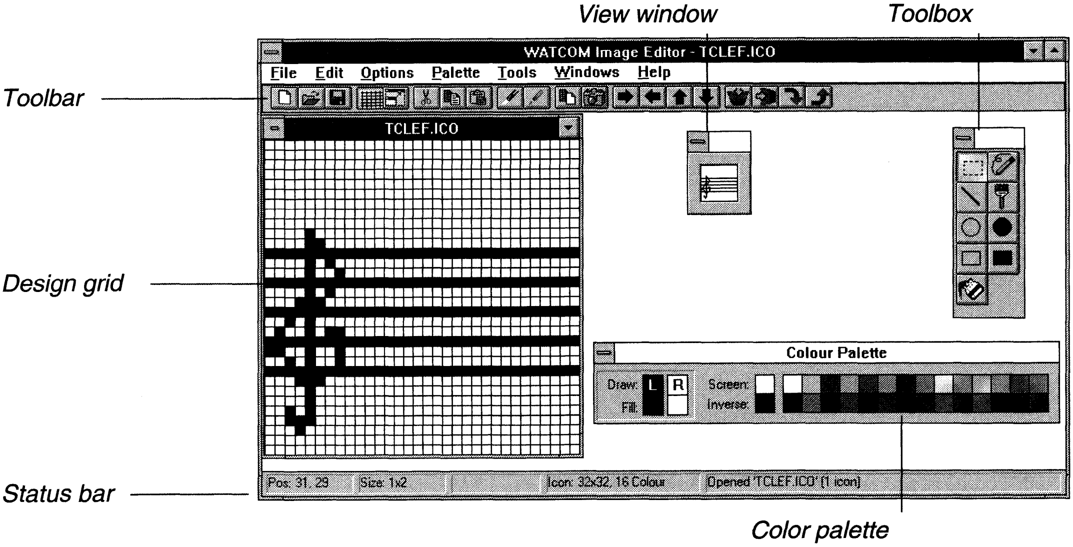
Exiting the Image Editor

❖ To exit the Image Editor, do one of the following:

- ♦ Press ALT+F4.
- ♦ Select File➤Exit from the menu bar.
- ♦ Double-click the System menu in the top-left corner of the screen.

Image Editor workspace

When you open the Image Editor, the main workspace appears. The toolbar, toolbox (also called tool window), and color palette appear in this window; the status bar appears at the bottom of the window.



Pos: 31, 29 Size: 1x2 Icon: 32x32, 16 Colour Opened 'TCLEF.ICO' [1 icon]

Creating a design grid

You cannot draw or copy images onto the screen itself. Instead, you must first create a **design grid**, a work area used to design visual images. You can create an empty design grid and draw your own visual image or open an existing bitmap, icon, or cursor. In addition to the design grid, you can display the **view window** to see the visual image in its real size.

❖ To create a design grid:

1 Select File➤New from the menu bar.

The Image Type dialog box displays.

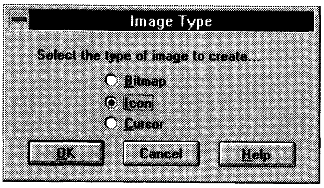


Image Type

Select the type of image to create...

○ Bitmap
● Icon
○ Cursor

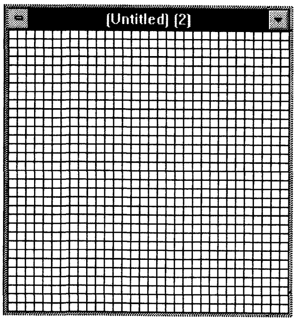
2 Select the type of image you want to create by clicking the Bitmap, Icon, or Cursor radio button.

3 Click OK.

A dialog box appears in which you specify the desired image size. The Image Editor automatically defaults to the standard sizes (in pixels) for bitmaps, icons, and cursors. To create different sized images, enter the appropriate pixel width and height.

4 Click OK.

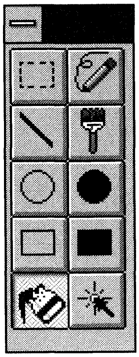
The design grid displays.



For more information on creating and editing bitmaps, cursors, and icons, see "Creating and editing bitmaps," "Creating and editing cursors," and "Creating and editing icons" later in this chapter.

The toolbox

The toolbox provides easy access to the tools you use to draw visual images.



❖ To use a tool:

1 Click on its icon.

2 Move the cursor over the design grid to the desired spot.

3 Click (or click and drag).

For example, to use the pencil tool, first click its icon and then click (or click and drag) to fill pixels on the design grid.

You can also enable tools by selecting them from the Tools menu.

☞ For complete information on Image Editor tools, see "Tools menu" on page 88.

Creating and editing bitmaps

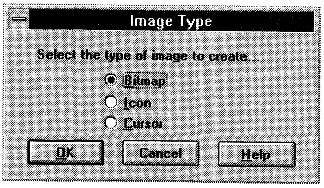
If you want to add visual impact to your applications, you can use **bitmaps** created and edited using the Image Editor. Bitmaps are graphical images that can be as large as 512 x 512 square pixels representing such things as buttons, brush patterns, company logos, toolbar items, wallpaper, and graphical elements in menus.

Using the Image Editor you can create a new bitmap or edit an existing one. You can even create a bitmap from another application on your desktop.

❖ To create a new bitmap:

1 Select File➤New from the menu bar.

The Image Type dialog box displays.

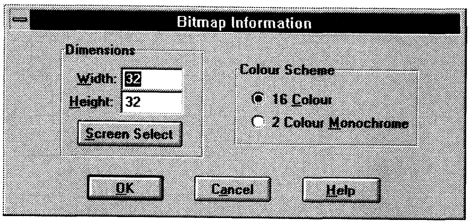


Select the type of image to create...
● Bitmap
○ Icon
○ Cursor

2 Click the Bitmap radio button.

3 Click OK.

The Bitmap Information dialog box displays.



Width: 32
Height: 32
● 16 Colour
○ 2 Colour Monochrome

This dialog box prompts you for information about the dimensions of the bitmap you want to create and the color scheme you want to work with.

The Image Editor defaults to the dimensions most commonly used for creating bitmaps, 32 pixels by 32 pixels. However, if you want to create a different sized bitmap, enter another pixel width and height.

The Image Editor also defaults to a 16-color color scheme. If you want to design a bitmap for monochrome monitors, select 2 Colour Monochrome.

4 Click OK.

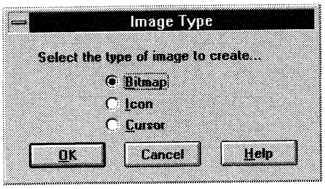
The design grid displays.

5 Edit and save the visual image.

❖ To copy a bitmap from your desktop or from another application on your desktop:

1 Select File➤New from the menu bar.

The Image Type dialog box displays.

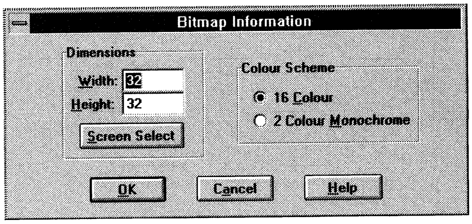


● Bitmap
○ Icon
○ Cursor

2 Select the Bitmap radio button.

3 Click OK.

The Bitmap Information dialog box displays.



Width: 32
Height: 32

● 16 Colour
○ 2 Colour Monochrome

4 Click the Screen Select button.

The Image Editor displays the Windows desktop.

This option allows you to choose the size and contents of the newly created bitmap with images from your desktop or from another application sitting on your desktop. The Image Editor ignores the limitations specified in the Bitmap Information dialog box and uses the dimensions you indicate through mouse actions described in the next three steps.

5 Position the cursor at the image you want to select.

6 Press and hold the left mouse button and drag the cursor across the image you want to copy to your design grid.

7 Release the mouse button.

The image you selected appears on the design grid.

8 Edit and save the visual image.

An alternative method
You can also use Snap Bitmap to copy from the Windows desktop into the Image Editor.

For information about Snap Bitmap, see "Copying images" on page 76.

Creating and editing cursors

You can create and edit **cursors** that help communicate to your users the mode or state of the application. For example, you might create a DataWindow that displays a customized I-beam cursor over text fields and a pointer over all other fields.

You can create a new cursor or copy an image from the desktop or another application onto your design grid. Then you can edit the cursor's appearance.

When you edit a cursor, you can set its **hot spot**, which determines the x,y coordinates that are indicated when the user presses the mouse button. The tool allows you to select the exact pixel to which the hot spot is assigned. The default hot spot is 0,0 (the top left corner of the design grid).

Cursors (and icons) differ from bitmaps in that they almost always include pixels that blend into the background of your application. Unless you want a rectangular-shaped cursor, you must assign special screen and inverse colors to the "blending pixels," so they don't obscure the areas over which you move the cursor.

Recommendation
Since you can only create black and white cursors, do not use black or white as your screen or inverse colors.

For more information on selecting screen and inverse colors, see "Selecting screen and inverse colors" on page 76.

❖ To create a new cursor:

1 Select File➤New from the menu bar.

The Image Type dialog box displays.

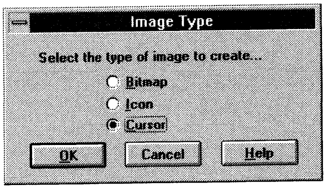


Image Type

Select the type of image to create...

○ Bitmap

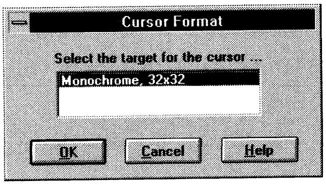
○ Icon

● Cursor

2 Click the Cursor radio button.

3 Click OK.

The Cursor Format dialog box displays.



Cursor Format

Select the target for the cursor ...

Monochrome, 32x32

Since cursors should be standardized, you can only choose monochrome cursors 32x32 pixels. You cannot create a colored cursor.

4 Click OK.

5 Edit and save the visual image.

❖ To specify the cursor's hot spot:

1 Create or edit the cursor, using the design grid.

2 Click the Set Hot Spot button on the toolbar.
or
Select Tools➤Hot Spot from the menu bar.

3 Position the cursor over the desired hot spot (note that the cursor changes shape once you enter the design grid) and click.

The status bar displays coordinates
Use the status bar to follow the exact coordinates before clicking to note the hot spot.

Creating and editing icons

With the Image Editor, you can create **icons** that look exactly the way you want. Icons are usually the user's first exposure to your application, so you want them to be as professional and meaningful as possible.

You can create a new icon or you can create an empty icon design grid and use the Image Editor's Snap Bitmap option to copy an image from the desktop to the design grid, where you can edit it as desired.

❖ To create a new icon:

1 Select File➤New from the menu bar.

The Image Type dialog box displays.

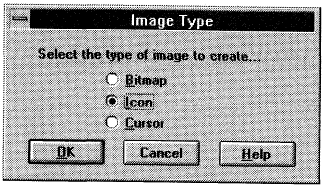


Image Type

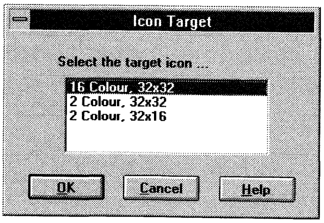
Select the type of image to create...

○ Bitmap
● Icon
○ Cursor

2 Click the Icon radio button.

3 Click OK.

The Icon Target dialog box displays. You can create a 16 color 32x32 icon, a 2 color 32x32 icon, or a 2 color 32x16 icon.



Icon Target

Select the target icon ...

16 Colour, 32x32
2 Colour, 32x32
2 Colour, 32x16

4 Click the desired icon type.

5 Click OK.

6 Edit and save the visual image.

Copying images

You can use the Image Editor's Snap Bitmap action to copy visual images from your Windows desktop or from other open Windows applications. Snap Bitmap can be used with bitmaps, icons, and cursors.

❖ To copy an image from your desktop or from another open application:

1 Establish a design grid for the bitmap, icon, or cursor.

2 Click the Snap Bitmap button on the toolbar.
or
Select Edit➤Snap Bitmap from the menu bar.

3 Position the cursor (which is now a gray rectangle) over the image to be copied and click.

The Image Editor copies the selected image to the design grid.

Selecting screen and inverse colors

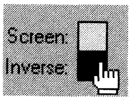
Unlike bitmaps, icons and cursors usually require transparent or inverse colors. This is so that they don't entirely obscure everything underneath them and so they can be seen no matter what color they happen to be sitting on.

When you create or edit an icon or cursor, you can choose the screen and inverse colors from the color palette. Then, you can draw portions of your icon or cursor with those colors, so that these pixels blend into the background of your application.

Remember, if you want to use a specific color in your icon or cursor, make sure the color you want to use is not the screen or inverse color. Reserve these two colors for drawing only the transparent portions of your icon or cursor and use the other 26 colors available in the color palette for regular drawing.

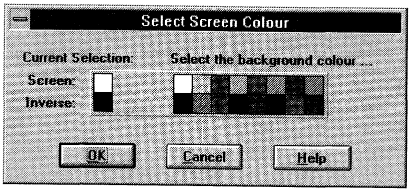
❖ To choose a screen color:

1. Create a design grid for a cursor or icon.
2. In the color palette, move your cursor over the two squares beside the words screen and inverse, so that your cursor changes from a pointer to a hand.



3. Double-click in either square.

   The Select Screen Colour dialog box displays.



4. Click the color you want to use as the screen color.
5. Click OK.

   The dialog box closes and the screen color and its inverse are assigned to the screen and inverse boxes in the color palette. These are the Screen and Inverse colors you can use to draw the portions of your icon or cursor that you want to be transparent.

About inverse colors

When you select a color, its inverse is automatically selected for you. Try to find a screen color and inverse color that you won't want to use in the foreground of your icon or cursor. For example, if you don't intend to use yellow in your icon, select yellow as your screen color. The inverse color, blue, is automatically selected for you; if you intend to use blue in your icon, use a different screen color that doesn't have blue as its inverse.

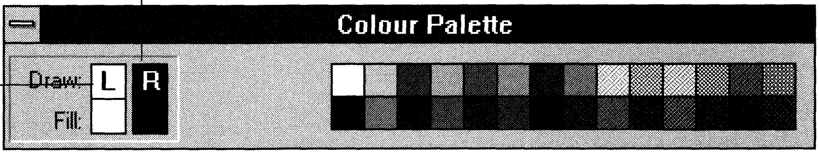
Assigning colors to mouse buttons

When you are drawing and filling in multiple colors, you may find it useful to assign separate colors to the left and right mouse buttons. This way, you can draw in one color, and quickly switch to another color simply by pressing the opposite mouse button. For example, if you find you draw mostly in black and red, assign black to the left mouse button and red to the right mouse button.

❖ To assign colors to mouse buttons:

1. In the color palette, select the left mouse button color by clicking the desired color with the left mouse button.
2. Select the right mouse button color by clicking the desired color with the right mouse button.

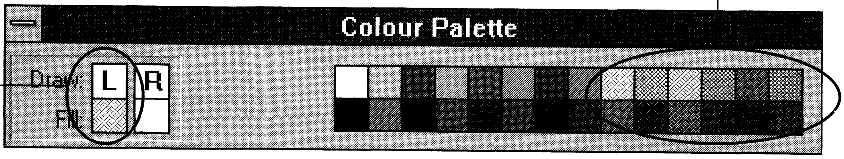
   The colors assigned to the mouse buttons appear in the squares of the color palette marked L (for left) and R (for right).



Using the color palette

You use the color palette to assign a color to the various drawing tools. For example, if you click the color red and then click the Pencil tool, the Image Editor draws pixels in red.

The 12 colors on the right side of the default color palette use nonsolid (or **dithered**) colors instead of solid colors. Windows creates dithered colors by alternating pixels of two or more solid colors so that, while the color appears solid in the view window, it is actually made up of different colored pixels.



Dithered colors

Draw and Fill colors differ for dithered colors

Since it is impossible to represent a single pixel-unit of a dithered color, when you select a dithered color, the Image Editor always substitutes a solid color for drawing (Pencil, Line, and Paintbrush tools) and uses the dithered color for Fill (solid Ellipse, solid Rectangle, and Fill tools).

❖ To select a color:

1 Click either the left or right mouse button on the color in the color palette that you wish to assign to that button.

2 Select a drawing tool.

3 Draw or fill holding down the mouse button assigned to the color you want.

For example, if you want to draw a red line:

1 Create a design grid for the desired visual image.

2 Select a shade of red from the color palette.

3 Select the line tool from the toolbox.

4 Click and drag your cursor across the design grid.

❖ To hide the color palette, use one of these methods.

- ♦ Double-click the System menu icon in the corner of the color palette.
- ♦ Toggle color palette display by selecting Options➤Show Colour Palette from the menu bar.
- ♦ Press F7.

❖ To reveal the color palette, use one of these methods.

- ♦ Toggle color palette display by selecting Options➤Show Colour Palette from the menu bar.
- ♦ Press F7.

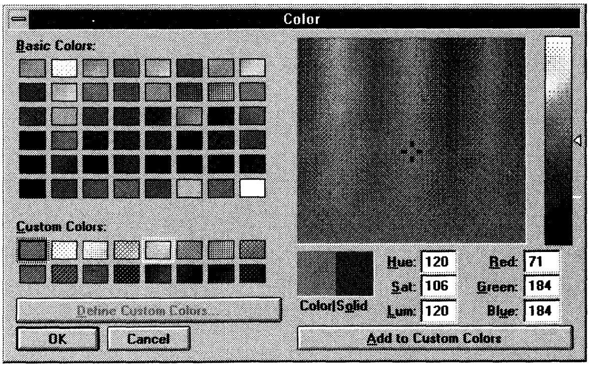
Changing colors

There may be times when you want to use a color other than the ones displayed on the color palette. The Image Editor allows you to replace colors on the color palette with other colors, which you specify through the Windows Color dialog box. You can save customized color palettes for use in later sessions.

❖ To change a color:

1 In the color palette, move the cursor over the color to be replaced with a custom color so that the cursor changes from a pointer to a hand and double-click.

The Windows Color dialog box displays.



Hue: 120 Red: 71
Sat: 106 Green: 184
Lum: 120 Blue: 184

Tip
You can also display this dialog box by selecting Palette➢Edit Current Colour from the Image Editor menu bar.

2 Choose an existing color or create the color you want. You can start with one of the basic colors and customize it in the palette to the right by dragging the color indicator with the mouse. You can also specify precise values to define the color.

3 When you have the color you want, click Add to Custom Colours.

The new color displays in the list of custom colors.

4 Select the new color in the list of custom colors.

5 Click OK.

The Image Editor redisplays with the color you specified replacing the original color.

❖ To save a customized color palette:

1. Select Palette➤Save Colour Palette from the menu bar.
2. Use the drive and directory dropdown list boxes to select the directory that will contain the customized palette file. Name the file using the PAL extension.
3. Click OK.

❖ To load a customized color palette:

1. Select Palette➤Load Colour Palette from the menu bar.
2. Use the drive and directory dropdown listboxes to select the directory containing the desired palette file.
3. Select the desired file and click OK.

Tip
You can revert to the default color palette by selecting Palette➤Restore Colour Palette from the menu bar.

Customizing current settings

You can use the Watcom Image Editor Current Settings dialog box to control certain aspects of Image Editor behavior.

- **Profile information** Whether current settings and window positions are saved after each session
- **Window display** The number of view windows to be displayed and whether the design grid is square by default
- **Shifting defaults** Whether to perform a wrapping shift or a clipping shift and the number of pixels to shift each time
- **Image pasting default** Whether to stretch a pasted image into the design grid or clip the pasted image

- **Image rotation defaults** Whether to perform clipping when rotated images don't fit (simple rotation will also clip) and whether to leave the rotated area selected

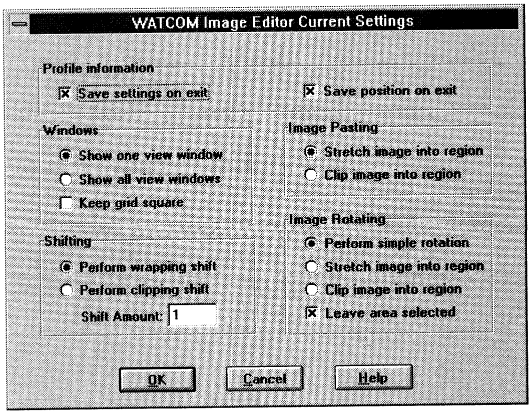
Settings are saved in the WATCOM.INI file

The Image Editor saves this information in the WATCOM.INI file, found in your Windows directory.

❖ To customize settings:

1 Select Options➤Current Settings from the menu bar.

The Watcom Image Editor Current Settings dialog box displays.



WATCOM Image Editor Current Settings

Profile information

☒ Save settings on exit

☒ Save position on exit

Windows

● Show one view window

○ Show all view windows

□ Keep grid square

Image Pasting

● Stretch image into region

○ Clip image into region

Shifting

● Perform wrapping shift

○ Perform clipping shift

Shift Amount: 1

Image Rotating

● Perform simple rotation

○ Stretch image into region

○ Clip image into region

☒ Leave area selected

2 Customize the desired Image Editor settings.

3 Click OK.

Image Editor menu bar

The Image Editor contains a menu bar, a toolbar, a toolbox, and a color palette:

- **The menu bar** along the top of the Image Editor window allows you to access most of the items in the toolbar, the toolbox, and the color palette. It also provides you with other items that you cannot access anywhere else.
- **The toolbar** allows you quick and easy access to the most used items in the menus.
- **The toolbox** allows you to conveniently select and change the tool you are using to edit your image, without having to hunt through the menus for the tool you want.

> **Tip**
> To see what an item does, position your cursor over the item and press the mouse button. A description of the item appears in the status line. If you only wish to read what the button does and not activate it, drag your cursor onto another part of the window before releasing the mouse button.

- **The color palette** allows you to specify the color used to draw or fill a new visual. You also use it to change the colors of an existing bitmap, icon, or cursor.

This section describes the menu bar and corresponding toolbar and toolbox options in more detail.

File menu

The File menu contains choices that retrieve, save, and close graphic files and windows.

Menu item	Activity	Shortcut key	Toolbar button
New	Opens a new window to begin work on a graphic image.	F2	🗋

Menu item	Activity	Shortcut key	Toolbar button
Open	Displays the Open Image File dialog box, allowing you to open an existing bitmap, icon, or cursor. Bitmaps have the BMP extension, icons have the ICO extension, and cursors have the CUR extension.	SHIFT+F2	📂
Save	Saves the current visual image.	F3	💾
Save As	Displays the Save Image File dialog box, allowing you to save a new visual image.	SHIFT+F3	
Close	Closes the current visual image.	SHIFT+F10	
Close All	Closes all open visual images.	CTRL+F10	
Exit	Exits the Image Editor.	ALT+F4	

Edit menu

The Edit menu contains choices that control the current visual image.

Menu item	Activity	Shortcut key	Toolbar button
Undo	Undoes previous changes to the visual image.	CTRL+U	↶
Redo	Redoes a previously undone action.	CTRL+R	↷
Restore	Restores the visual image to the last saved version.		
Cut	Cuts the selected area to the clipboard, removing it from the visual image.	CTRL+X	✂
Copy	Copies the selected area to the clipboard.	CTRL+C	📄
Paste	Pastes clipboard contents into the current visual image.	CTRL+V	📋

Menu item	Activity	Shortcut key	Toolbar button
Snap Bitmap	Allows you to select a bitmap image from your desktop or another application and place that image on the design grid		📷
Clear	Clears the current visual image, displaying a blank design grid.	DELETE	🗋
Shift➤Right	Shifts the visual image right, wrapping or clipping one pixel at a time, depending on the Current Settings specification.	SHIFT+ RIGHT ARROW	➡
Shift➤Left	Shifts the visual image left, wrapping or clipping one pixel at a time, depending on the Current Settings specification.	SHIFT+ LEFT ARROW	⬅
Shift➤Up	Shifts the visual image up, wrapping or clipping one pixel at a time, depending on the Current Settings specification.	SHIFT+ UP ARROW	⬆
Shift➤Down	Shifts the visual image down, wrapping or clipping one pixel at a time, depending on the Current Settings specification.	SHIFT+ DOWN ARROW	⬇
Flip➤Over Horizontal Axis	Flips the visual image horizontally.		⇄
Flip➤Over Vertical Axis	Flips the visual image vertically.		⇅
Rotate➤Counter Clockwise	Rotates the visual image 90º to the right, wrapping or clipping, depending on the Current Settings specification.		↺
Rotate➤ Clockwise	Rotates the visual image 90º to the left, wrapping or clipping, depending on the Current Settings specification.		↻

Menu item	Activity	Shortcut key	Toolbar button
Add an Icon	Displays the Icon Target dialog box, allowing you to add a new size icon to the current icon (ICO) file. An icon file can contain up to three separate images: one for each icon size (32x32 16-color, 32x32 2-color, and 32x16 2-color). Windows automatically selects the appropriate icon, based on the display type.		
Select Icon Image	Displays the Icon Target dialog box, allowing you to select one of the images from the current icon file.		
Delete an Icon	Displays the Icon Target dialog box, allowing you to delete one of the images from the current icon file.		

Options menu

The Options menu contains choices that control how your application looks and acts.

Menu item	Activity	Shortcut key	Toolbar button
Current Settings	Displays the Watcom Image Editor Current Settings dialog box, allowing you to specify profile, pasting, shifting, and rotation defaults	CTRL+S	
Grid	Toggles display of the grid	CTRL+G	▦
Keep Grid Square	Constrains the design grid to always resize to a square, depending on the Current Settings specification	CTRL+K	

Menu item	Activity	Shortcut key	Toolbar button
Maximize Current Image	Enlarges the design grid to fill up the entire window	CTRL+M	▣
Brush Size	Specifies the number of pixels painted by the brush tool, up to 5 pixels	CTRL+2, 3, 4, or 5	
Change Image Size	Displays the Change Image Size dialog box, allowing you to specify a new height and width		
Show Tool Window	Toggles display of the tool window (toolbox)	F6	
Show Colour Palette	Toggles display of the color palette	F7	
Show View Windows	Toggles display of the view window	F8	

Palette menu

The Palette menu contains choices that control the current color, the screen color, and the color palette.

Menu item	Activity
Edit Current Colour	Displays the Color dialog box, allowing you to replace the current color with a customized color. You can also do this by double-clicking the current color.
Reset Current Palette	Replaces all customized colors with those of the current palette.
Set Screen Colour	Displays the Select Screen Colour dialog box, allowing you to specify the screen color for icons and cursors.
Load Colour Palette	Displays the Open Colour Palette File dialog box, allowing you to load a customized color palette.
Save Colour Palette	Displays the Save Colour Palette File dialog box, allowing you to save the current palette for use at a later time.
Restore Colour Palette	Reloads the default color palette.

Tools menu

The Tools menu contains items that enable Image Editor tools.

Menu item	Activity	Toolbar button
Pencil	Enables the Pencil tool, which draws on the grid, pixel by pixel	✏
Line	Enables the Line tool, which draws straight lines on the grid	╲
Ellipse➤Outline	Enables the Ellipse Outline tool, which draws outlines of circles and ellipses	○
Ellipse➤Solid	Enables the Ellipse Solid tool, which draws solid circles and ellipses	●
Rectangle➤Outline	Enables the Rectangle Outline tool, which draws outlines of rectangles	□
Rectangle➤Solid	Enables the Rectangle Solid tool, which draws solid rectangles	■
Fill	Enables the Fill tool, which fills the indicated area with the current color	🪣
Brush	Enables the Brush tool, which draws on the grid in the current brush size	🖌
Select Region	Enables the Select Region tool, which selects a rectangular region on the design grid	⬚
Hot Spot	Enables the Hot Spot tool, which allows you to specify the cursor hot spot (for cursors only)	✛

Windows menu

The Windows menu contains standard choices for altering window placement.

Menu item	Activity	Shortcut key
Arrange Icons	Arranges all minimized visual images along the bottom of the window	
Tile Windows	Arranges and sizes all current windows such that they don't overlap	SHIFT+F4
Cascade Windows	Causes all open windows to cascade from the upper left across to the lower right	SHIFT+F5

Help menu

The Help menu contains standard choices for online product information.

Menu item	Activity
About	Displays the About Watcom Image Editor dialog box
Help	Displays online Help for the Image Editor

CHAPTER 7

Stored Procedure Update Utility

About this chapter

This chapter describes the Stored Procedure Update utility.

Contents

Stored Procedure Update overview

The Stored Procedure Update utility generates PowerScript statements that you can use to override default DataWindow behavior and update the database through stored procedures.

Your DBMS must support stored procedures
To use this utility, your DBMS must support stored procedures.

About Stored Procedure Update

Database stored procedures allow you to define procedural SQL statements in the database for use by all applications. Using stored procedures to perform database updates allows you to enhance database security, integrity, and performance. Since stored procedures provide for conditional execution, you can also use them to enforce additional business rules.

You implement database update through stored procedures by:

1. Revoking UPDATE, INSERT, and DELETE rights from all users.
2. Selectively granting stored procedure EXECUTE rights to the appropriate users as stored procedures are created.

Although DataWindows can use stored procedures as a retrieval data source, the DataWindow Update function always updates the database by dynamically generating INSERT, DELETE, and UPDATE SQL statements. This can be an issue when standards requires that stored procedures perform all database updates.

PowerBuilder provides the SQLPreview event, which is invoked just before it submits a SQL statement to the database. This event allows you to override the default DataWindow update capability by creating a script that updates the database through a stored procedure.

The Stored Procedure Update utility automatically creates PowerScript statements that you can use in a DataWindow's SQLPreview event to perform database updates through stored procedures.

How Stored Procedure Update works

Stored Procedure Update generates PowerScript statements based on information that you provide. These statements override default DataWindow database update processing and invoke your stored procedures instead. You then paste or import the PowerScript statements into your DataWindow's SQLPreview event.

You can use Stored Procedure Update to invoke stored procedures in either of two ways:

- Name stored procedures directly.
- Name PowerBuilder user object functions that use remote procedure calls (RPCs) to invoke database stored procedures. Such a user object must be a Standard Class user object of type Transaction.

  ☞ For more information on implementing remote procedure calls in user objects, see *Building Applications* in the PowerBuilder documentation set.

Recommendations

- Don't create stored procedures until all DataWindows have been tested completely. This will save you the trouble of revising stored procedures due to DataWindow design changes
- Your stored procedures should contain WHERE clauses that name all columns in the table. This will enhance database integrity by preventing lost updates.

Error handling assumptions

Stored Procedure Update generates PowerScript code that checks the transaction object's SQLCODE attribute to determine whether the database update succeeded. Your stored procedure or user object function must use SQLCODE to indicate success or failure. You can do this by using a database function such as Sybase's RAISERROR or by setting the SQLCODE attribute explicitly in the user object function.

Using Stored Procedure Update

This section describes how to use the Stored Procedure Update utility:

- Starting Stored Procedure Update
- Exiting Stored Procedure Update
- Stored Procedure Update workspace
- Selecting an application
- Selecting a DataWindow
- Specifying update criteria
- Updating the DataWindow's SQLPreview event
- Specifying options

Starting Stored Procedure Update

❖ To start Stored Procedure Update, do one of the following.

- Double-click the Stored Procedure Update utility icon from the Windows Program Manager.
- Double-click SPUD.EXE in the Windows File Manager.
- Select File➤Run from the Windows Program Manager menu bar and type:

  ***directory name*\spud**

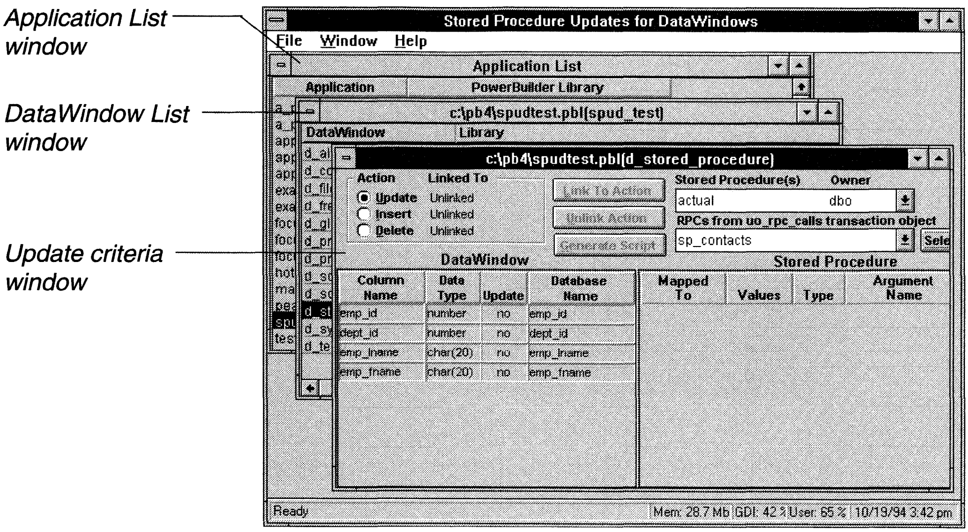
Exiting Stored Procedure Update

❖ To exit Stored Procedure Update, do one of the following.

- Press ALT+F4.
- Select File➤Exit from the menu bar.
- Double-click the System menu in the top-left corner of the window.

Stored Procedure Update workspace

Here is the Stored Procedure Update workspace.



Action	Linked To
● Update	Unlinked
○ Insert	Unlinked
○ Delete	Unlinked

Stored Procedure(s): actual   Owner: dbo

RPCs from uo_rpc_calls transaction object: sp_contacts

Column Name	Data Type	Update	Database Name
emp_id	number	no	emp_id
dept_id	number	no	dept_id
emp_lname	char(20)	no	emp_lname
emp_fname	char(20)	no	emp_fname

Mapped To	Values	Type	Argument Name

Ready   Mem: 28.7 Mb GDI: 42 % User: 65 % 10/19/94 3:42 pm

The Application List window lists the application objects and the PBL files in which they reside. The DataWindow List window lists all DataWindows in the selected application. The update criteria window contains fields used to create stored procedure update PowerScript statements.

If the Application List window is empty

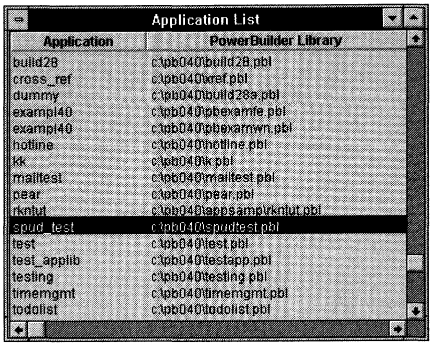
The Stored Procedure Update utility uses the PB.INI file to obtain the application list. If the PB.INI file cannot be found, a message displays with an empty Application List window. This problem may indicate that the Advanced PowerBuilder Utilities were not installed in the PowerBuilder directory.

If a "Cannot read from drive" message displays

The Stored Procedure Update utility checks to see if each application named in the PB.INI file exists. If this application resides on a drive that is unavailable (such as an empty diskette drive or an unattached network drive), file errors will interrupt the process. Click Cancel to continue.

Selecting an application

Once Stored Procedure Update is running, select one of your PowerBuilder applications from the Application List window. This window lists all of the PowerBuilder application objects and the PBLs in which each object resides.



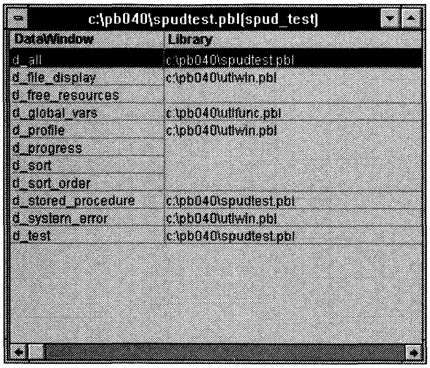
Application	PowerBuilder Library
build28	c:\pb040\build28.pbl
cross_ref	c:\pb040\xref.pbl
dummy	c:\pb040\build28a.pbl
exampl40	c:\pb040\pbexamfe.pbl
exampl40	c:\pb040\pbexamwn.pbl
hotline	c:\pb040\hotline.pbl
kk	c:\pb040\k.pbl
mailtest	c:\pb040\mailtest.pbl
pear	c:\pb040\pear.pbl
rkntut	c:\pb040\appsamp\rkntut.pbl
spud_test	c:\pb040\spudtest.pbl
test	c:\pb040\test.pbl
test_applib	c:\pb040\testapp.pbl
testing	c:\pb040\testing.pbl
timemgmt	c:\pb040\timemgmt.pbl
todolist	c:\pb040\todolist.pbl

❖ To select an application from the Application List window, do one of the following.

- ♦ Double-click on the application.
- ♦ Highlight an application row using the UP ARROW or DOWN ARROW keys, then press ENTER.

Selecting a DataWindow

After you have selected an application, you can select the DataWindow for which Stored Procedure Update will generate PowerScript statements. You do this using the DataWindow selection window.



DataWindow	Library
d_all	c:\pb040\spudtest.pbl
d_file_display	c:\pb040\utlwin.pbl
d_free_resources	
d_global_vars	c:\pb040\utlfunc.pbl
d_profile	c:\pb040\utlwin.pbl
d_progress	
d_sort	
d_sort_order	
d_stored_procedure	c:\pb040\spudtest.pbl
d_system_error	c:\pb040\utlwin.pbl
d_test	c:\pb040\spudtest.pbl

❖ To select a DataWindow, do one of the following:

- ♦ Double-click on the DataWindow name.
- ♦ Highlight a DataWindow row using the UP ARROW or DOWN ARROW keys, then press ENTER.

Specifying update criteria

You use the update criteria window to:

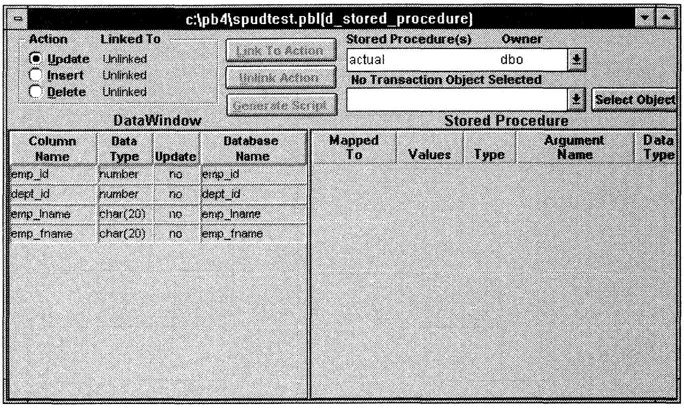
- ♦ Select a database update action (UPDATE, INSERT, DELETE)
- ♦ Select a user object containing functions that perform remote procedure calls (optional)
- ♦ Select a stored procedure or remote procedure call that performs the specified action

> **Stored procedures and remote procedure calls**
> All references to stored procedures apply equally to remote procedure calls.

- ♦ Map DataWindow fields to stored procedure arguments

- ♦ Link the database update action to the stored procedure using the specified mapping
- ♦ Generate PowerScript statements that enable the DataWindow to update the database through stored procedures

A sample window is shown below.


c:\pb4\spudtest.pbl(d_stored_procedure)

Action	Linked To
● Update	Unlinked
○ Insert	Unlinked
○ Delete	Unlinked

Stored Procedure(s): actual Owner: dbo

No Transaction Object Selected

DataWindow

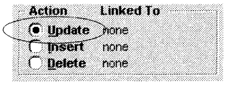
Column Name	Data Type	Update	Database Name
emp_id	number	no	emp_id
dept_id	number	no	dept_id
emp_lname	char(20)	no	emp_lname
emp_fname	char(20)	no	emp_fname

Stored Procedure

Mapped To	Values	Type	Argument Name	Data Type

❖ To generate PowerScript statements that enable the DataWindow to update the database through stored procedures:

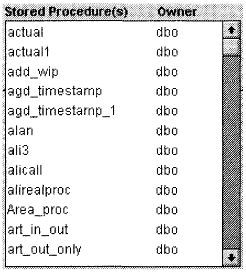
1 Select an action by clicking the desired radio button in the Action box.



Action	Linked To
● Update	none
○ Insert	none
○ Delete	none

2 Click Select Stored Procedure.

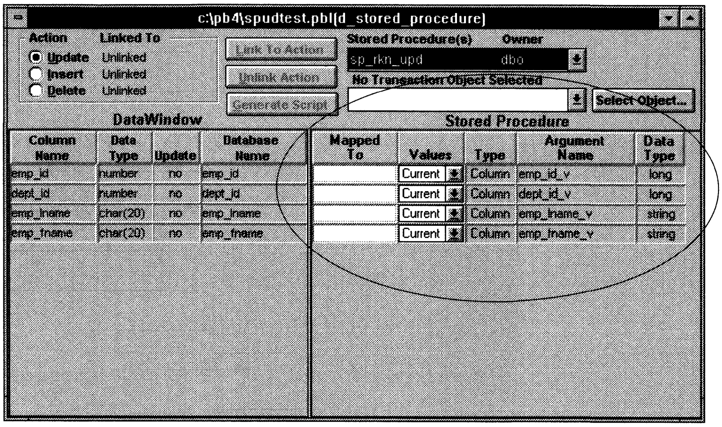
A dropdown listbox displays naming all stored procedures for the current database.



Stored Procedure(s)	Owner
actual	dbo
actual1	dbo
add_wip	dbo
agd_timestamp	dbo
agd_timestamp_1	dbo
alan	dbo
ali3	dbo
alicall	dbo
alirealproc	dbo
Area_proc	dbo
art_in_out	dbo
art_out_only	dbo

3 Scroll through the list and click on the stored procedure you want to use for the action selected in step 1.

The stored procedure's arguments display in the Stored Procedure box.



c:\pb4\spudtest.pbl(d_stored_procedure)

Action	Linked To
● Update	Unlinked
○ Insert	Unlinked
○ Delete	Unlinked

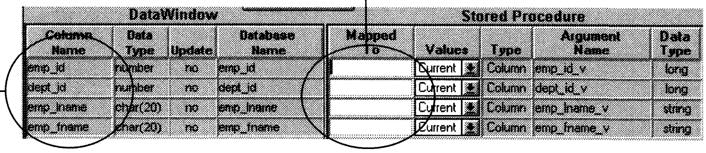
Stored Procedure(s): sp_rkn_upd Owner: dbo

No Transaction Object Selected

DataWindow				Stored Procedure				
Column Name	Data Type	Update	Database Name	Mapped To	Values	Type	Argument Name	Data Type
emp_id	number	no	emp_id		Current	Column	emp_id_v	long
dept_id	number	no	dept_id		Current	Column	dept_id_v	long
emp_lname	char(20)	no	emp_lname		Current	Column	emp_lname_v	string
emp_fname	char(20)	no	emp_fname		Current	Column	emp_fname_v	string

4 Map DataWindow fields to stored procedure arguments by dragging DataWindow fields and dropping them over the associated stored procedure argument name. Stored Procedure Update will not allow data type mismatches.

To here



DataWindow				Stored Procedure				
Column Name	Data Type	Update	Database Name	Mapped To	Values	Type	Argument Name	Data Type
emp_id	number	no	emp_id		Current	Column	emp_id_v	long
dept_id	number	no	dept_id		Current	Column	dept_id_v	long
emp_lname	char(20)	no	emp_lname		Current	Column	emp_lname_v	string
emp_fname	char(20)	no	emp_fname		Current	Column	emp_fname_v	string

One-by-one, drag from here

You can also type directly into the Mapped To column. Stored Procedure Update uses the following rules to determine mapping type:

- If the value you type matches a column name, Stored Procedure Update uses the column value for argument mapping
- If the value you type is enclosed in quotation marks, Stored Procedure Update assumes a literal and uses the literal value for column mapping
- If the value you type has no quotes and the data type is not numeric, Stored Procedure Update assumes that it is a PowerScript variable, for which you will provide a value in the SQLPreview event

Stored Procedure Update passes NULL values for arguments for which no mapping is specified.

Original value versus Current value
The Values dropdown listbox displays two items: Current and Original. Current passes the current DataWindow column value to the stored procedure. Original passes the initial DataWindow column value. You should choose Initial if the column is used in a WHERE clause for UPDATE or DELETE.

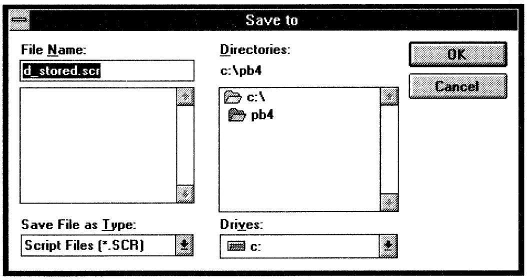
5 Repeat steps 1 through 4 for all appropriate actions.

6 Optionally modify the transaction object named in the PowerScript statements, as described in "Specifying options," on page 104.

7 Click Generate Script.

8 *If you did not link stored procedures to all three actions*, Stored Procedure Update displays a warning message box. Click OK.

If you specified that Stored Procedure Update should save PowerScript statements to a file, the Save to dialog box displays, prompting for a filename.



File Name: d_stored.scr

Directories: c:\pb4

Save File as Type: Script Files (*.SCR)

Drives: c:

9 Optionally, change the suggested filename and click OK.

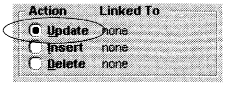
Stored Procedure Update creates PowerScript statements that use the specified stored procedures to update the database and writes them to the selected destinations (clipboard and/or file), as specified in the Options dialog box.

❖ To generate PowerScript statements based on remote procedure calls encapsulated in a Standard Class user object of type Transaction:

1 Select a DataWindow to be updated using stored procedures.

2 Modify the transaction object named in the PowerScript statements, as described in "Specifying options," on page 104. If using a nonstandard transaction object for SQLCA, you can still use SQLCA here.

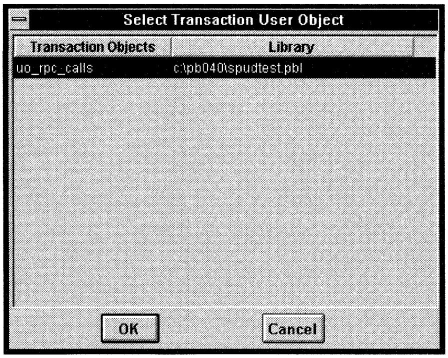
3 Select an action by clicking the desired radio button in the Action box.



Action	Linked To
● Update	none
○ Insert	none
○ Delete	none

4 Click Select Object.

The Select Transaction User Object dialog box displays all user objects of type Transaction.



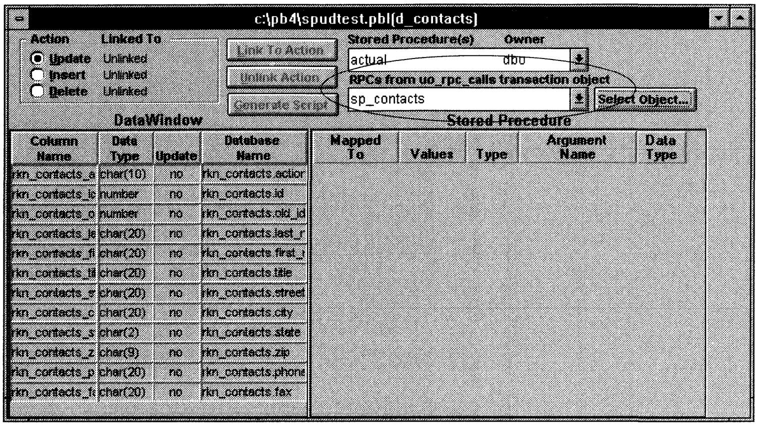
Select Transaction User Object

Transaction Objects	Library
uo_rpc_calls	c:\pb040\spudtest.pbl

5 Select the user object containing remote procedure calls to update your DataWindow.

6 Click OK.

The update criteria window displays. The RPCs from transaction object dropdown listbox is now enabled.



Action / Linked To
● Update Unlinked
○ Insert Unlinked
○ Delete Unlinked

Stored Procedure(s): actual Owner: dbo

RPCs from uo_rpc_calls transaction object: sp_contacts

DataWindow

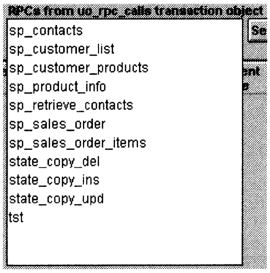
Column Name	Data Type	Update	Database Name
rkn_contacts_a	char(10)	no	rkn_contacts.actior
rkn_contacts_ic	number	no	rkn_contacts.id
rkn_contacts_o	number	no	rkn_contacts.old_id
rkn_contacts_le	char(20)	no	rkn_contacts.last_r
rkn_contacts_fi	char(20)	no	rkn_contacts.first_i
rkn_contacts_ti	char(20)	no	rkn_contacts.title
rkn_contacts_s	char(20)	no	rkn_contacts.street
rkn_contacts_c	char(20)	no	rkn_contacts.city
rkn_contacts_s	char(2)	no	rkn_contacts.state
rkn_contacts_z	char(9)	no	rkn_contacts.zip
rkn_contacts_p	char(20)	no	rkn_contacts.phone
rkn_contacts_fa	char(20)	no	rkn_contacts.fax

Stored Procedure

Mapped To	Values	Type	Argument Name	Data Type

7 Click the down arrow next to the RPCs from transaction object dropdown listbox.

A dropdown list displays all functions in the selected user object.

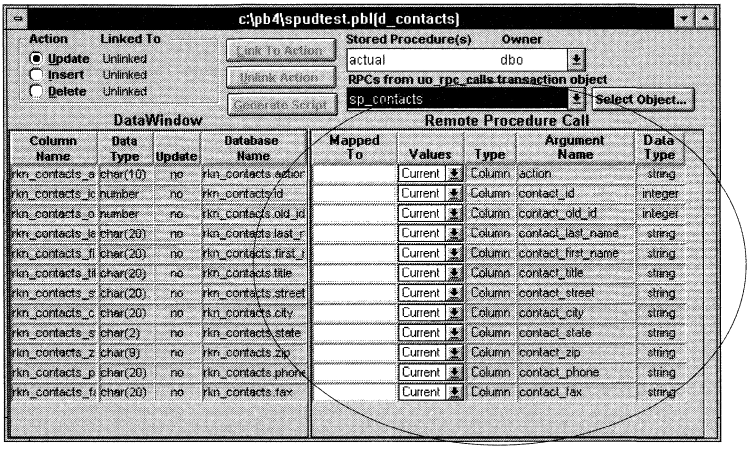


RPCs from uo_rpc_calls transaction object

sp_contacts
sp_customer_list
sp_customer_products
sp_product_info
sp_retrieve_contacts
sp_sales_order
sp_sales_order_items
state_copy_del
state_copy_ins
state_copy_upd
tst

8 Scroll through the list and click on the function that executes the stored procedure for the action selected in step 3.

The stored procedure's arguments display in the Remote Procedure Call box.



Action	Linked To
● Update	Unlinked
○ Insert	Unlinked
○ Delete	Unlinked

Stored Procedure(s): actual Owner: dbo

RPCs from uo_rpc_calls transaction object: sp_contacts

Column Name	Data Type	Update	Database Name	Mapped To	Values	Type	Argument Name	Data Type
rkn_contacts_a	char(10)	no	rkn_contacts.actior		Current	Column	action	string
rkn_contacts_ic	number	no	rkn_contacts.id		Current	Column	contact_id	integer
rkn_contacts_o	number	no	rkn_contacts.old_id		Current	Column	contact_old_id	integer
rkn_contacts_la	char(20)	no	rkn_contacts.last_r		Current	Column	contact_last_name	string
rkn_contacts_fi	char(20)	no	rkn_contacts.first_r		Current	Column	contact_first_name	string
rkn_contacts_til	char(20)	no	rkn_contacts.title		Current	Column	contact_title	string
rkn_contacts_s	char(20)	no	rkn_contacts.street		Current	Column	contact_street	string
rkn_contacts_c	char(20)	no	rkn_contacts.city		Current	Column	contact_city	string
rkn_contacts_s	char(2)	no	rkn_contacts.state		Current	Column	contact_state	string
rkn_contacts_z	char(9)	no	rkn_contacts.zip		Current	Column	contact_zip	string
rkn_contacts_p	char(20)	no	rkn_contacts.phone		Current	Column	contact_phone	string
rkn_contacts_fa	char(20)	no	rkn_contacts.fax		Current	Column	contact_fax	string

9 Continue as described in steps 4 through 9 of the previous procedure.

Updating the DataWindow's SQLPreview event

You use the PowerScript painter to paste or insert the PowerScript statements that Stored Procedure Update generates into the DataWindow control's SQLPreview event.

❖ To update the DataWindow control's SQLPreview event:

1 Open the Window painter and select the window that contains the DataWindow to be updated with the generated PowerScript statements.

2 Display the PowerScript painter for the DataWindow control.

3 Select the SQLPreview event.

4 Select Edit➤Paste from the menu bar (if statements were generated to the clipboard).
or
Select File➤Import from the menu bar and select the generated script file (if statements were generated to a file).

PowerBuilder inserts the generated PowerScript statements.

5 Review the PowerScript statements and modify them, as needed.

Specifying options

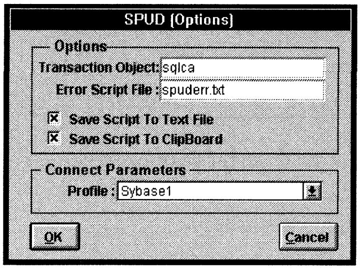
You can adjust the default settings for the following:

- Transaction object named in statements generated by Stored Procedure Update
- Error file used as a template for error handling statements generated by Stored Procedure Update
- Whether to copy PowerScript statements to the Windows clipboard
- Whether to create a file with the PowerScript statements
- Connection profile

❖ To specify Stored Procedure Update options:

1 Select File➤Options from the menu bar.

The Options dialog box displays.



SPUD (Options)

Options

Transaction Object: sqlca

Error Script File : spuderr.txt

☒ Save Script To Text File

☒ Save Script To ClipBoard

Connect Parameters

Profile : Sybase1

2 Update options as desired:

- **Use the Options section** to specify the transaction object that Stored Procedure Update names in the PowerScript statements that it generates. You also specify the error filename, as well as whether to generate PowerScript statements to the clipboard, to a text file, or both.
- **Use the Connect Parameters section** to select a database connection profile as defined in the PB.INI file.

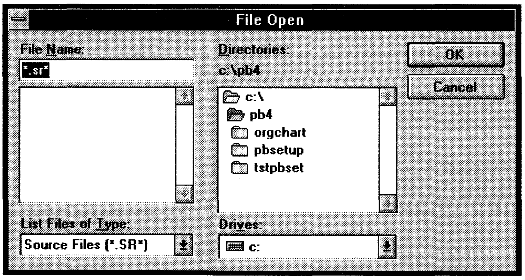
3 Click OK.

Stored Procedure Update saves the options in the SPUD.INI file.

❖ To modify error handling statements:

1 If you are in PowerBuilder, access the PowerBuilder File Editor by pressing SHIFT+F6 (otherwise, use the Windows Notepad or some other ASCII text editor).

The File Open dialog box displays.



File Name: *.sr*

Directories: c:\pb4

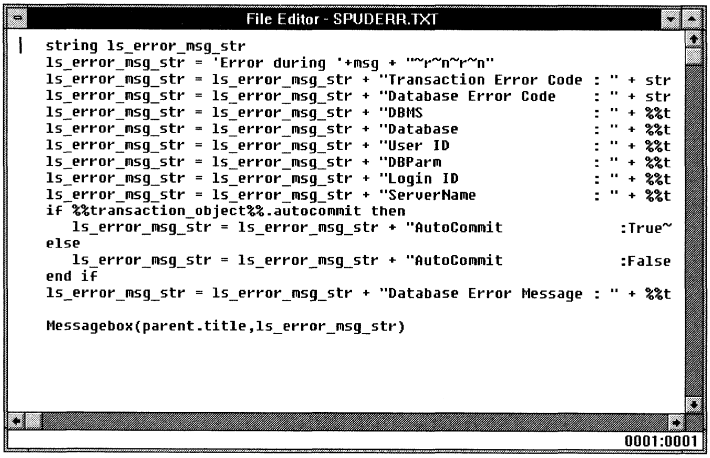
c:\
pb4
orgchart
pbsetup
tstpbset

List Files of Type: Source Files (*.SR*)

Drives: c:

2 Select the SPUDERR.TXT file and Click OK.

The File Editor displays the requested file.



```
string ls_error_msg_str
ls_error_msg_str = 'Error during '+msg + "~r~n~r~n"
ls_error_msg_str = ls_error_msg_str + "Transaction Error Code : " + str
ls_error_msg_str = ls_error_msg_str + "Database Error Code    : " + str
ls_error_msg_str = ls_error_msg_str + "DBMS                   : " + %%t
ls_error_msg_str = ls_error_msg_str + "Database               : " + %%t
ls_error_msg_str = ls_error_msg_str + "User ID                : " + %%t
ls_error_msg_str = ls_error_msg_str + "DBParm                 : " + %%t
ls_error_msg_str = ls_error_msg_str + "Login ID               : " + %%t
ls_error_msg_str = ls_error_msg_str + "ServerName             : " + %%t
if %%transaction_object%%.autocommit then
   ls_error_msg_str = ls_error_msg_str + "AutoCommit             :True~
else
   ls_error_msg_str = ls_error_msg_str + "AutoCommit             :False
end if
ls_error_msg_str = ls_error_msg_str + "Database Error Message : " + %%t

Messagebox(parent.title,ls_error_msg_str)
```

3 Review and modify error handling statements, as appropriate.

Use %%transaction_object%% for flexibility

For statements and functions that require the transaction object as a qualifier or parameter, you can type **%%transaction_object%%** and the Stored Procedure Update utility will substitute at script creation time with the appropriate transaction object name, as specified in the Options section.

4 Save the changes by selecting File➤Save from the menu bar.

5 Exit the PowerBuilder File Editor by selecting File➤Exit from the menu bar.

Stored Procedure Update menu bar

The Stored Procedure Update utility contains a menu bar. This section describes the menu items unique to Stored Procedure Update.

☞ For information on the Window and Help menus, see the discussions in Chapter 2, "Cross Reference."

File menu

The File menu contains choices that perform actions on Stored Procedure Update windows.

Menu item	Activity	Shortcut key
Options	Displays the Options dialog box, which allows you to specify defaults for transaction object, error file, script file, and profile	
Close	Closes the current window. You cannot close the Application List window	CTRL+F4
Exit	Closes Stored Procedure Update	ALT+F4

CHAPTER 8
Install Diskette Builder

About this chapter

This chapter describes the Install Diskette Builder.

Contents

Install Diskette Builder overview

The Install Diskette Builder allows you to create installation diskettes for PowerBuilder applications.

About the Install Diskette Builder

Deploying an application can be a major project all by itself. The Install Diskette Builder can help. It allows you to create installation diskettes that contain compressed application files and runtime components.

Users install the diskettes you create by running the Microsoft Windows Setup program. That is, you can deploy applications that use the same installation process as PowerBuilder and other Microsoft Windows applications.

How the Install Diskette Builder works

The Install Diskette Builder creates diskette image directories and installation diskettes from predefined component lists and reusable components. You use the Install Diskette Builder to define:

- **An installation configuration (CFG) file** In a configuration file, you list application components, the files used by each component, and INI file settings.
- **Reusable components** You can define reusable components to copy to diskette image files (PowerBuilder and DBMS executable files are the most common uses for reusable components)
- **A Program Group** You can specify that the setup process will create a Program Group and you can name the Program Items displayed in the Program Group.
- **Diskette images** You can create diskette image files on your hard drive (or on a network drive).
- **Installation diskettes** You use the Install Diskette Builder to copy from diskette image files to blank diskettes. The user installs from diskette in the usual way.

Using the Install Diskette Builder

This section describes how to use the Install Diskette Builder.

- Starting the Install Diskette Builder
- Exiting the Install Diskette Builder
- Install Diskette Builder workspace
- Defining components
- Defining reusable components
- Defining a Program Group
- Using a PowerBuilder project object
- Creating diskette images
- Creating diskettes
- Installing from diskettes

Starting the Install Diskette Builder

❖ **To start the Install Diskette Builder, do one of the following:**

- Double-click the Install Diskette Builder icon from the Windows Program Manager.
- Double-click PBSETUP.EXE in the Windows File Manager.
- Select File➢Run from the Windows Program Manager menu bar and type:

  directory name**\pbsetup**

The Install Diskette Builder banner displays.

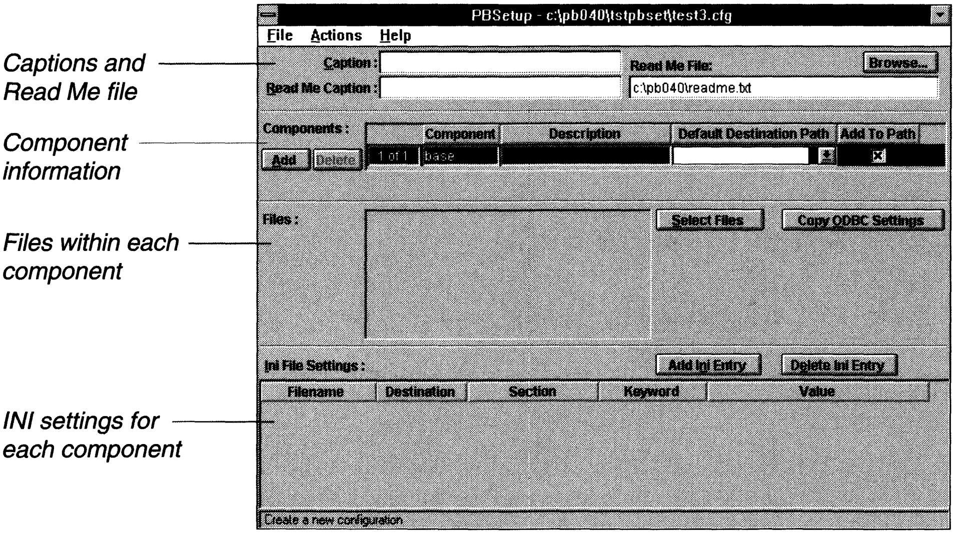
Exiting the Install Diskette Builder

❖ To exit the Install Diskette Builder, do one of the following:

- ♦ Press ALT+F4.
- ♦ Select File➣Exit from the menu bar.
- ♦ Double-click the system menu in the top, left corner of the window.

Install Diskette Builder workspace

The Install Diskette Builder workspace is shown below.



Captions and Read Me file

Component information

Files within each component

INI settings for each component

PBSetup - c:\pb040\tstpbset\test3.cfg

Caption :

Read Me Caption :

Read Me File: c:\pb040\readme.txt

Components :

	Component	Description	Default Destination Path	Add To Path
1 of 1	base			☒

Files :

Ini File Settings :

Filename	Destination	Section	Keyword	Value

Create a new configuration

Defining components

You use the Install Diskette Builder to define installation components. Each component contains settings, specifications, and a list of related files. The Install Diskette Builder uses the components that you define to create diskette image directories, from which you can create installation diskettes.

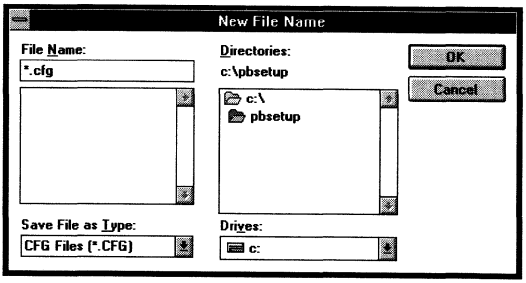
The Install Diskette Builder stores installation information in a configuration (CFG) file. The information in this file includes:

- Headings used by the Windows setup program
- The installation components
- The files within each component
- Default path and directory names for each component
- Whether to add a component's directory to the user's AUTOEXEC.BAT file
- INI file settings, including ODBC.INI settings for ODBC data sources

❖ To define a configuration file:

1 Select File➤New from the menu bar.

The New File Name dialog box displays.



File Name: *.cfg

Directories: c:\pbsetup

c:\
pbsetup

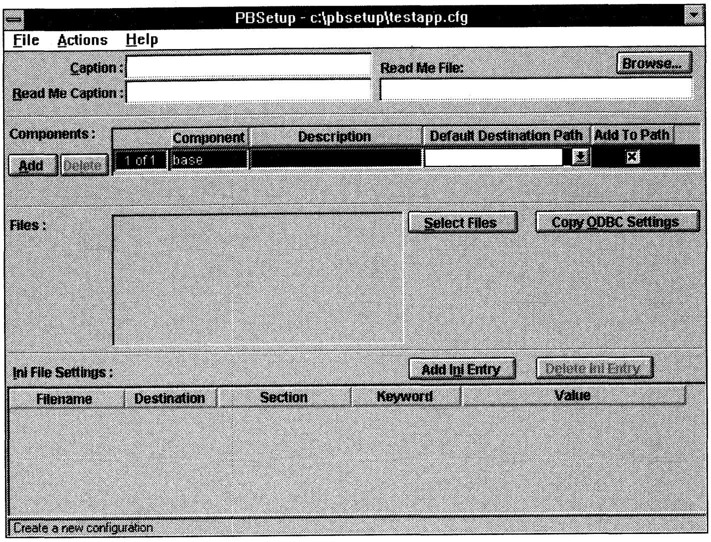
Save File as Type: CFG Files (*.CFG)

Drives: c:

2 Change the drive and directory, as needed, and specify a name for the configuration file in the File Name box.

3 Click OK.

The Install Diskette Builder workspace displays.



PBSetup - c:\pbsetup\testapp.cfg

Component: 1 of 1 base

Create a new configuration

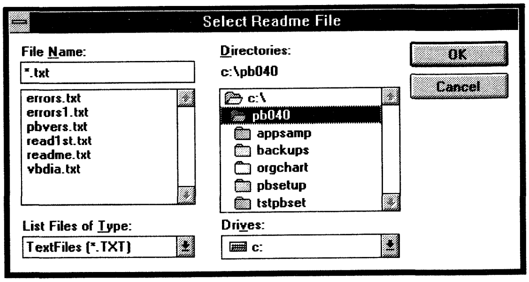
❖ To define window titles and a Read Me file:

1 Specify a title for the Install window in the Caption box.

2 Specify a title for the Read Me window in the Read Me Caption box.

> **The Read Me window**
> The install process displays the Read Me window while copying files. You can use it to highlight application-specific issues.

3 Specify the name of the text file to be displayed in the Read Me window. If you don't know the full path and name of the Read Me file, click Browse, which displays the Select Readme File dialog box.



File Name: *.txt

errors.txt
errors1.txt
pbvers.txt
read1st.txt
readme.txt
vbdia.txt

Directories: c:\pb040

c:\
pb040
appsamp
backups
orgchart
pbsetup
tstpbset

List Files of Type: TextFiles (*.TXT)

Drives: c:

4 Change the drive and directory, as needed, and select the text file to be displayed in the Read Me window.

5 Click OK.

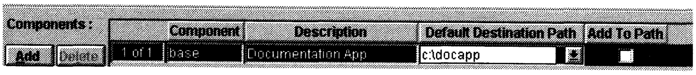
❖ To define a component and component files:

1 Click in the Description box and type a short component description.

2 Click in the Default Destination Path box and type a default pathname for the files in this component.

> **Specifying a relative pathname**
> Click on the down arrow in the Default Destination Path dropdown listbox to place component files in the WINDOWS or WINDOWS\SYSTEM directory. For components other than the Base, you can also specify that component files will be installed in the Base directory.

3 If desired, select the Add to Path checkbox to add the component path to the AUTOEXEC.BAT PATH statement.

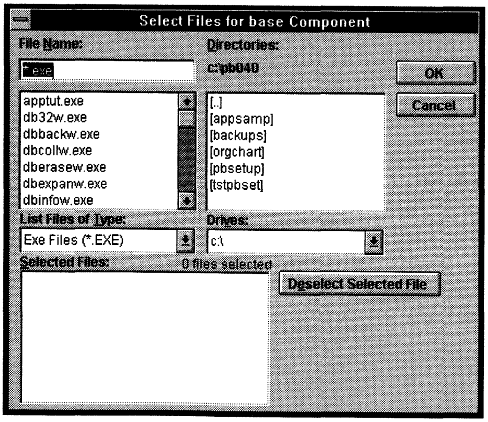


Components:

	Component	Description	Default Destination Path	Add To Path
1 of 1	base	Documentation App	c:\docapp	□

4 Click Select Files.

The Select Files for base Component dialog box displays.



Select Files for base Component

File Name: *.exe

Directories: c:\pb040

apptut.exe
db32w.exe
dbbackw.exe
dbcollw.exe
dberasew.exe
dbexpanw.exe
dbinfow.exe

[..]
[appsamp]
[backups]
[orgchart]
[pbsetup]
[tstpbset]

List Files of Type: Exe Files (*.EXE)

Drives: c:\

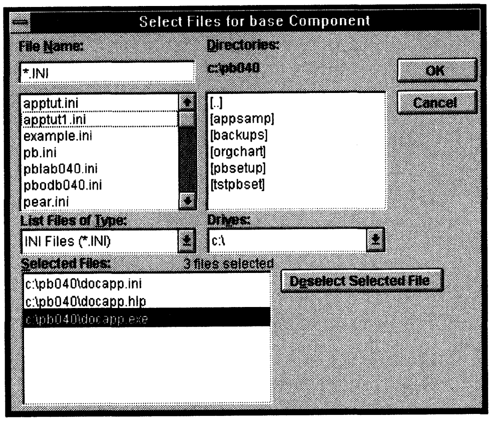
Selected Files: 0 files selected

5 Select drives and directories, as necessary to display the desired files. Use the List Files of Type dropdown listbox and the File Name box to select the files that are displayed.

6 Include a file by doing one of the following:

- Double-click the filename in the top box.
- Drag the file from the top box to the Selected Files box.

The Install Diskette Builder moves the file from the top box to the Selected Files box.



Select Files for base Component

File Name: *.INI

apptut.ini
apptut1.ini
example.ini
pb.ini
pblab040.ini
pbodb040.ini
pear.ini

Directories: c:\pb040

[..]
[appsamp]
[backups]
[orgchart]
[pbsetup]
[tstpbset]

List Files of Type: INI Files (*.INI)

Drives: c:\

Selected Files: 3 files selected

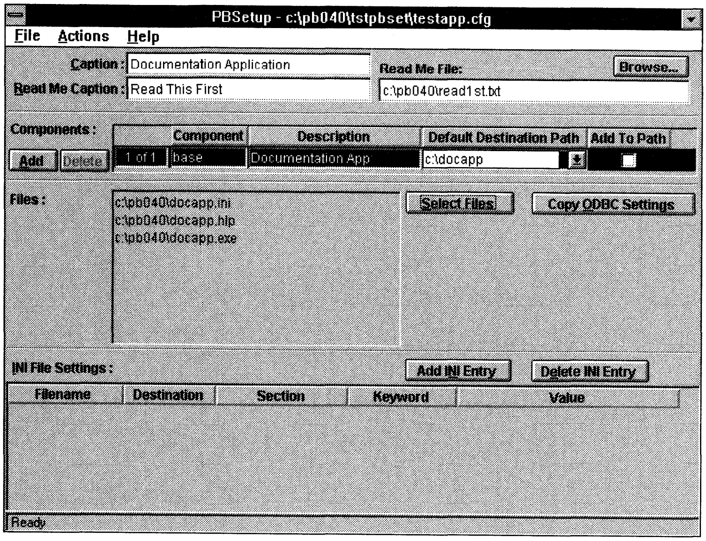
c:\pb040\docapp.ini
c:\pb040\docapp.hlp
c:\pb040\docapp.exe

7 Remove a file from the Selected Files list by doing one of the following:

- Click the filename, then click Deselect Selected File.
- Drag the file from the Selected Files box to the top box.

8 When you have selected all files for the component, click OK.

The Install Diskette Builder main workspace displays with the selected file names.



PBSetup - c:\pb040\tstpbset\testapp.cfg

Caption : Documentation Application

Read Me Caption : Read This First

Read Me File: c:\pb040\read1st.txt

Components :

	Component	Description	Default Destination Path	Add To Path
1 of 1	base	Documentation App	c:\docapp	□

Files :

c:\pb040\docapp.ini
c:\pb040\docapp.hlp
c:\pb040\docapp.exe

INI File Settings :

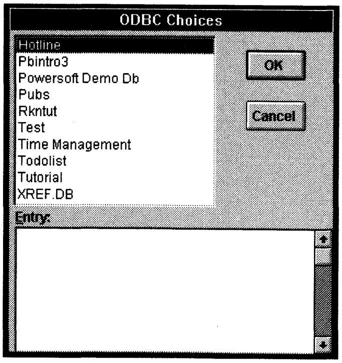
Filename	Destination	Section	Keyword	Value

Ready

❖ To copy ODBC.INI settings to the INI File Settings box:

1 Click Copy ODBC Settings.

The ODBC Choices dialog box displays.



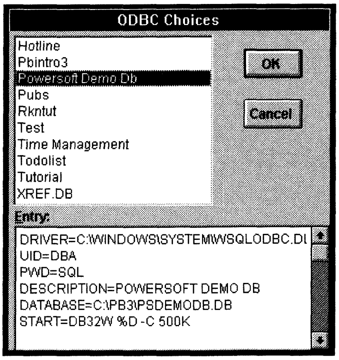
ODBC Choices

Hotline
Pbintro3
Powersoft Demo Db
Pubs
Rkntut
Test
Time Management
Todolist
Tutorial
XREF.DB

Entry:

2 Click on the data source to be included with the component.

The data source's individual ODBC.INI line entries display in the Entry box.



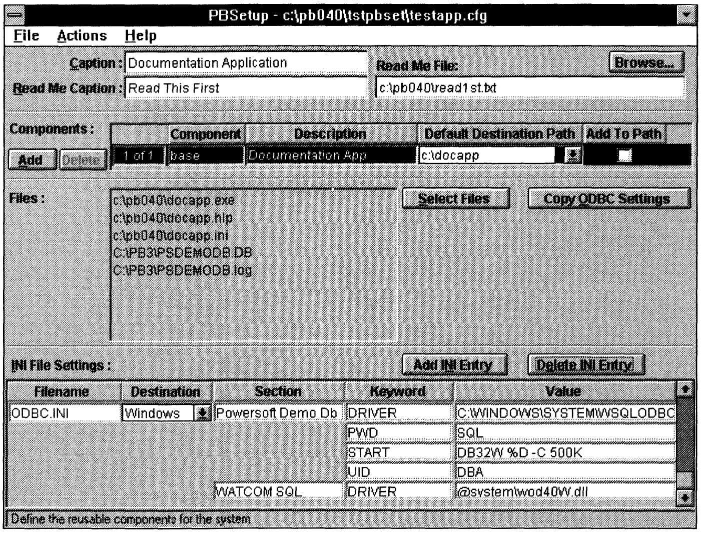
Hotline
Pbintro3
Powersoft Demo Db
Pubs
Rkntut
Test
Time Management
Todolist
Tutorial
XREF.DB

```
DRIVER=C:\WINDOWS\SYSTEM\WSQLODBC.Dl
UID=DBA
PWD=SQL
DESCRIPTION=POWERSOFT DEMO DB
DATABASE=C:\PB3\PSDEMODB.DB
START=DB32W %D -C 500K
```

3 Review the Entry lines to ensure the correct choice.

4 Click OK.

The Install Diskette Builder main workspace displays with the INI file settings filled in. At install time, the install process will copy these settings to the user's ODBC.INI file.



Caption: Documentation Application

Read Me Caption: Read This First

Read Me File: c:\pb040\read1st.txt

Components:

	Component	Description	Default Destination Path	Add To Path
1 of 1	base	Documentation App	c:\docapp	□

Files:

c:\pb040\docapp.exe
c:\pb040\docapp.hlp
c:\pb040\docapp.ini
C:\PB3\PSDEMODB.DB
C:\PB3\PSDEMODB.log

INI File Settings:

Filename	Destination	Section	Keyword	Value
ODBC.INI	Windows	Powersoft Demo Db	DRIVER	C:\WINDOWS\SYSTEM\WSQLODBC
			PWD	SQL
			START	DB32W %D -C 500K
			UID	DBA
		WATCOM SQL	DRIVER	@system\wod40W.dll

Define the reusable components for the system

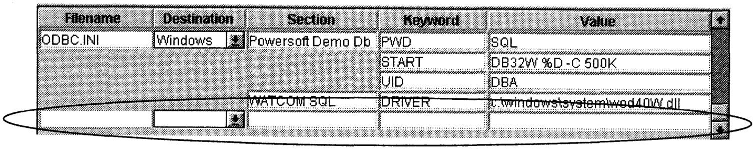
❖ To add INI file settings:

1 Click an existing entry on the level at which you want to add a new entry.

For example, if you want to add an entry to another file, click an entry in the Filename column; if you want to add an entry to another section within a file that is already displayed, click an entry in the Section column.

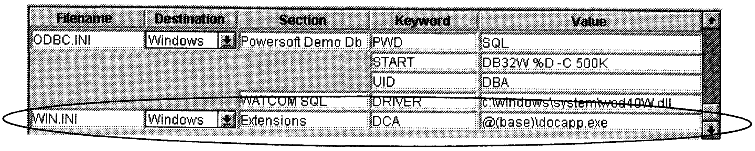
2 Click Add INI Entry.

The Install Diskette Builder displays an empty line.



Filename	Destination	Section	Keyword	Value
ODBC.INI	Windows	Powersoft Demo Db	PWD	SQL
			START	DB32W %D -C 500K
			UID	DBA
		WATCOM SQL	DRIVER	c:\windows\system\wod40W.dll

3 Specify filename, destination, section, keyword, and value, as appropriate.



Filename	Destination	Section	Keyword	Value
ODBC.INI	Windows	Powersoft Demo Db	PWD	SQL
			START	DB32W %D -C 500K
			UID	DBA
		WATCOM SQL	DRIVER	c:\windows\system\wod40W.dll
WIN.INI	Windows	Extensions	DCA	@(base)\docapp.exe

Using the @(base) variable

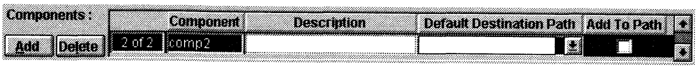
This example uses the @(base) variable as part of the value that specifies the pathname for DOCAPP.EXE. At install time, the installation program replaces @(base) with the pathname the user specifies for the base component, *not including the trailing backslash* (so you must include the backslash, as shown in this example).

In addition to @(base), you can also use @(windows) to specify the pathname of the WINDOWS directory, @(system) to specify the pathname of the WINDOWS\SYSTEM directory, and @(component) to specify the pathname the user specifies for a component directory.

❖ To define another component:

1 Click Add.

A blank component entry line displays.



	Component	Description	Default Destination Path	Add To Path
2 of 2	comp2			□

2 Add a description and path information, as appropriate.

3 Specify files.

4 Update INI settings, as appropriate.

❖ To save the configuration file:

- ♦ Select File➤Save from the menu bar.

  The Install Diskette Builder saves the updated configuration.

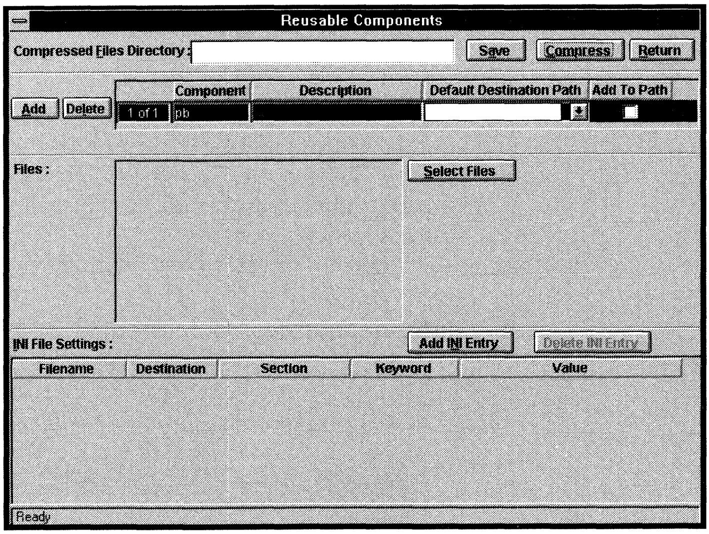
Defining reusable components

You typically define reusable components for PowerBuilder executable files and for your DBMS's executable files. This saves you the time it would take to define these files separately for each configuration file. Depending on your applications and operating environment, you may find many other uses for reusable components.

❖ To define reusable components:

1 Select Actions➤Define Reusable Components from the menu bar.

The Reusable Components window displays.

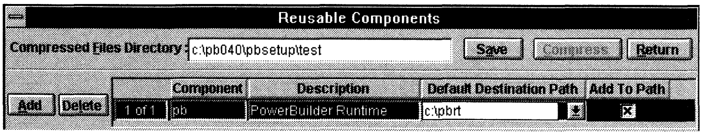


	Component	Description	Default Destination Path	Add To Path
1 of 1	pb			□

Ready

2 Name the directory to contain the compressed files for reusable components.

3 Specify a description, default installation directory for this component's files, and whether to add the component path to the AUTOEXEC.BAT PATH statement.

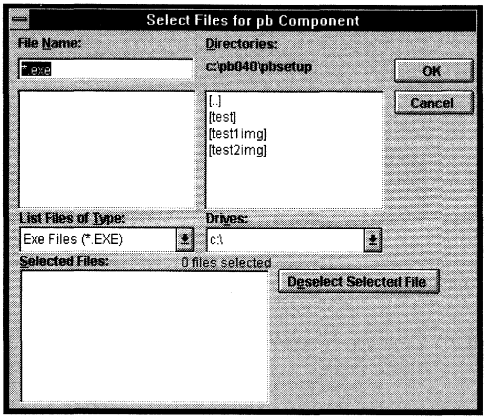

Reusable Components

Compressed Files Directory : c:\pb040\pbsetup\test

	Component	Description	Default Destination Path	Add To Path
1 of 1	pb	PowerBuilder Runtime	c:\pbrt	☒

4 Click Select Files.

The Select Files for pb Component dialog box displays.


Select Files for pb Component

File Name: *.exe

Directories: c:\pb040\pbsetup

[..]
[test]
[test1img]
[test2img]

List Files of Type: Exe Files (*.EXE)

Drives: c:\

Selected Files: 0 files selected

5 Select the drives and directories, as necessary to display the desired files. Use the List Files of Type dropdown listbox and the File Name box to select the files that are displayed.

6 Include a file by doing one of the following:

- ♦ Double-click the filename in the top box.
- ♦ Drag the file from the top box to the Selected Files box.

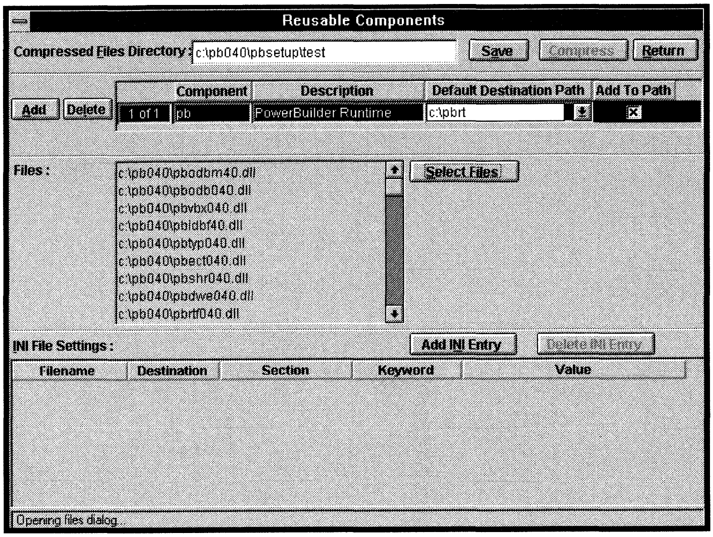
The Install Diskette Builder moves the file from the top box to the Selected Files box.

7 Remove a file from the Selected Files list by doing one of the following:

- ♦ Click the filename, then click Deselect Selected File.
- ♦ Drag the file from the Selected Files box to the top box.

8 When you have selected all files for the reusable component, click OK.

The Reusable Components window displays with the selected file names.



Compressed Files Directory: c:\pb040\pbsetup\test

	Component	Description	Default Destination Path	Add To Path
1 of 1	pb	PowerBuilder Runtime	c:\pbrt	☒

Files:

```
c:\pb040\pbodbm40.dll
c:\pb040\pbodb040.dll
c:\pb040\pbvbx040.dll
c:\pb040\pbidbf40.dll
c:\pb040\pbtyp040.dll
c:\pb040\pbect040.dll
c:\pb040\pbshr040.dll
c:\pb040\pbdwe040.dll
c:\pb040\pbrtf040.dll
```

INI File Settings:

Filename	Destination	Section	Keyword	Value

Opening files dialog...

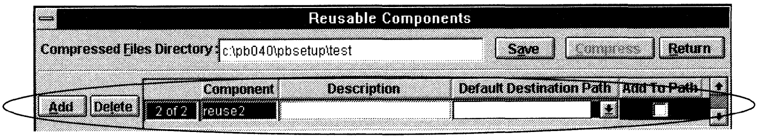
❖ To update INI file settings:

♦ To specify INI file settings for the reusable component, use the INI File Settings area (see the instructions on page 118).

❖ To define another reusable component:

1 Click Add.

A blank component entry line displays.



Compressed Files Directory: c:\pb040\pbsetup\test

	Component	Description	Default Destination Path	Add To Path
2 of 2	reuse2			□

2 Add a description and path information, as appropriate.

3 Specify files.

4 Update INI settings, as appropriate.

❖ To save the reusable components:

♦ Click Save.

The Install Diskette Builder saves your specifications in the PBSETUP.INI file.

You must save reusable components before you can create compressed versions.

❖ To create compressed versions of reusable modules:

♦ Click Compress.

The Install Diskette Builder creates compressed versions of all modules named in reusable component lists. It saves these compressed files in the directory named in the Compressed Files Directory box. The Install Diskette Builder uses these files when creating diskette image directories, as specified in the Create Diskette Images dialog box.

> **Compression is a DOS process**
> When you click Compress, the Install Diskette Builder executes a DOS process and displays a status window along with a minimized DOS window. Do not interrupt the compression process.

❖ To close the Reusable Components window:

♦ Click Return.

The Install Diskette Builder main workspace displays.

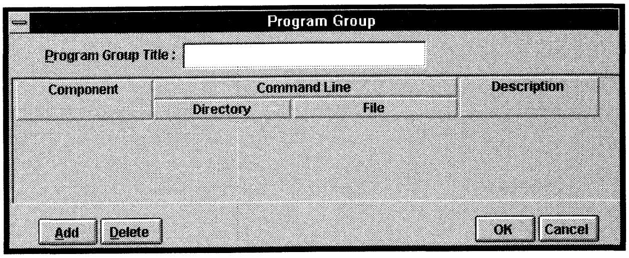
Defining a Program Group

You can specify that the installation process will create a Windows Program Group. You can also specify the files that display in the Program Group.

❖ To define a Program Group created by the installation process:

1 Select Actions➤Define Program Group from the menu bar.

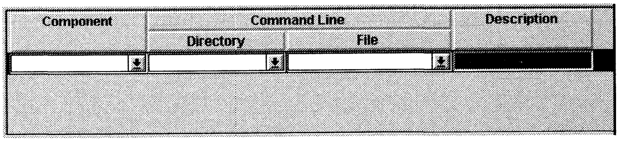
The Program Group dialog box displays.



2 Specify a Program Group title.

3 Click Add.

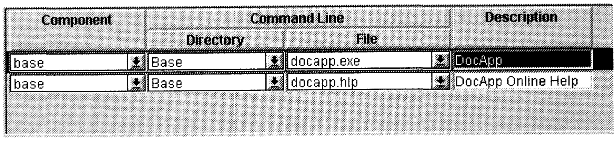
An entry line displays.



4 Specify the installation component, installation directory, filename, and a description.

Use the dropdown listboxes
Use the dropdown listboxes to save time and minimize entry errors.

5 Click Add to display another entry line and add another Program Item.



Component	Command Line		Description
	Directory	File	
base	Base	docapp.exe	DocApp
base	Base	docapp.hlp	DocApp Online Help

6 When you have added all desired Program Items to the Program Group, click OK.

The Install Diskette Builder main workspace displays.

7 Select File➤Save from the menu bar.

The Install Diskette Builder updates the configuration file with the Program Group specifications.

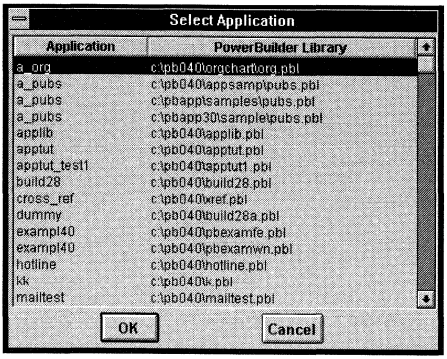
Using a PowerBuilder project object

You can use a PowerBuilder project object as the basis for a component's file list.

❖ To use a PowerBuilder project object to populate a component list:

1 Select Actions➤Read Project Object from the menu bar.

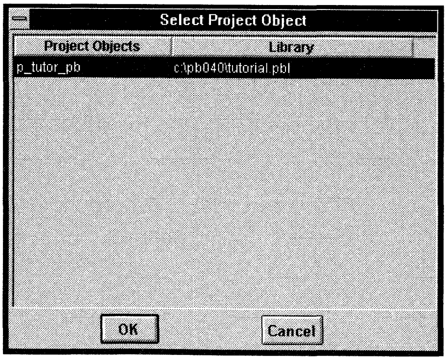
The Select Application dialog box displays.



Application	PowerBuilder Library
a_org	c:\pb040\orgchart\org.pbl
a_pubs	c:\pb040\appsamp\pubs.pbl
a_pubs	c:\pbapp\samples\pubs.pbl
a_pubs	c:\pbapp30\sample\pubs.pbl
applib	c:\pb040\applib.pbl
apptut	c:\pb040\apptut.pbl
apptut_test1	c:\pb040\apptut1.pbl
build28	c:\pb040\build28.pbl
cross_ref	c:\pb040\xref.pbl
dummy	c:\pb040\build28a.pbl
exampl40	c:\pb040\pbexamfe.pbl
exampl40	c:\pb040\pbexamwn.pbl
hotline	c:\pb040\hotline.pbl
kk	c:\pb040\k.pbl
mailtest	c:\pb040\mailtest.pbl

2 Double-click the application containing the desired project object.

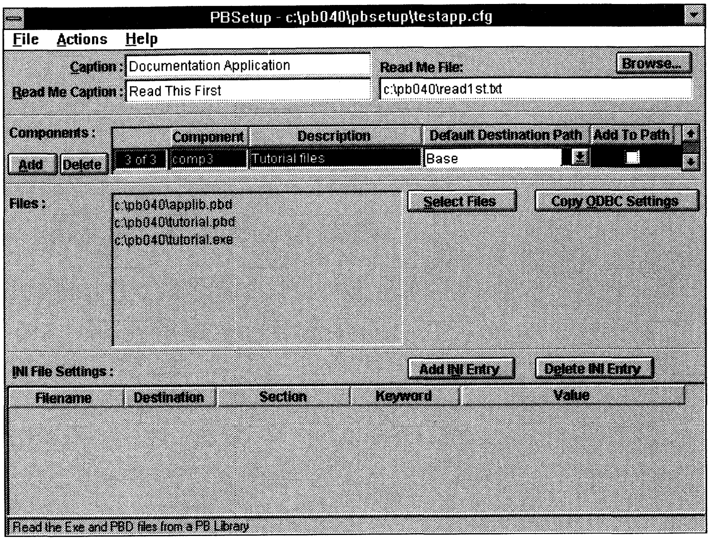
The Select Project Object dialog box displays.



Project Objects	Library
p_tutor_pb	c:\pb040\tutorial.pbl

3 Double-click the project object whose modules (EXE and PBD files) you want to include in the component.

The Install Diskette Builder main workspace displays with the EXE and PBD files named in the project object.



PBSetup - c:\pb040\pbsetup\testapp.cfg

Caption : Documentation Application

Read Me Caption : Read This First

Read Me File: c:\pb040\read1st.txt

Components :

	Component	Description	Default Destination Path	Add To Path
3 of 3	comp3	Tutorial files	Base	□

Files :

c:\pb040\applib.pbd
c:\pb040\tutorial.pbd
c:\pb040\tutorial.exe

INI File Settings :

Filename	Destination	Section	Keyword	Value

Read the Exe and PBD files from a PB Library

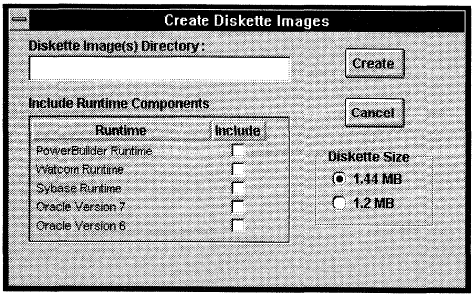
Creating diskette images

You use the Install Diskette Builder to create a directory structure of diskette images from which install diskettes are created.

❖ To create diskette images:

1. Display the configuration for which you want to create diskette images.
2. Select Actions➢Create Diskette Images from the menu bar. (The Create Diskette Images menu item is disabled until you save the configuration.)

The Create Diskette Images dialog box displays.



Create Diskette Images

Diskette Image(s) Directory:

Include Runtime Components

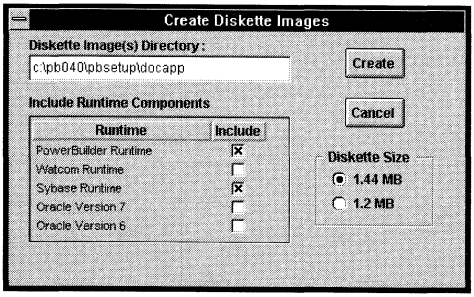
Runtime	Include
PowerBuilder Runtime	□
Watcom Runtime	□
Sybase Runtime	□
Oracle Version 7	□
Oracle Version 6	□

Diskette Size
● 1.44 MB
○ 1.2 MB

3 Specify the pathname of the directory to contain the diskette images.

4 Specify which runtime components to include.

5 Specify diskette size.



Create Diskette Images

Diskette Image(s) Directory: c:\pb040\pbsetup\docapp

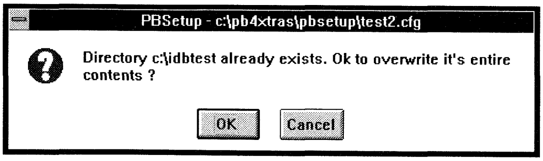
Include Runtime Components

Runtime	Include
PowerBuilder Runtime	☒
Watcom Runtime	□
Sybase Runtime	☒
Oracle Version 7	□
Oracle Version 6	□

Diskette Size
● 1.44 MB
○ 1.2 MB

6 Click Create.

7 If the directory you named already contains files, the Install Diskette Builder asks you whether to delete them.



PBSetup - c:\pb4xtras\pbsetup\test2.cfg

Directory c:\idbtest already exists. Ok to overwrite it's entire contents ?

8 Click OK.

The Install Diskette Builder creates directories, compresses files, and populates the directories with component files and reusable files.

The Install Diskette Builder creates subdirectories named DISK1, DISK2, DISK3, and so on.

Creating diskettes

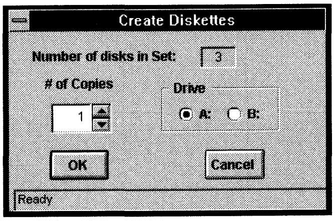
Once diskette images have been created, you can create installation diskettes at any time.

❖ To create diskettes:

1 Display the configuration for which you wish to create diskettes.

2 Select Actions➤Create Diskettes from the menu bar. (The Create Diskettes menu item is disabled until you create diskette images.)

The Create Diskettes dialog box displays.



Create Diskettes

Number of disks in Set: 3

# of Copies: 1

Drive: ● A: ○ B:

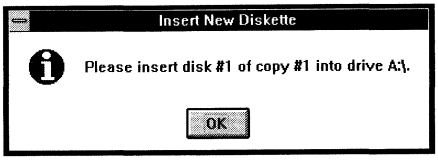
Ready

3 Specify the number of copies to make.

4 Specify which diskette drive to use.

5 Click OK.

The Insert New Diskette dialog box displays.

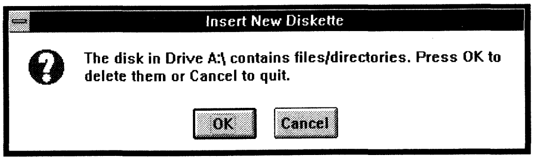


Insert New Diskette

Please insert disk #1 of copy #1 into drive A:\.

6 Click OK.

7 If the diskette already contains files, the Install Diskette Builder asks you whether to delete them.



Insert New Diskette

The disk in Drive A:\ contains files/directories. Press OK to delete them or Cancel to quit.

8 Click OK.

The Install Diskette Builder copies files to diskette, prompting you to add new diskettes, as needed.

When diskette copying is complete, the Install Diskette Builder main workspace displays.

Installing from diskettes

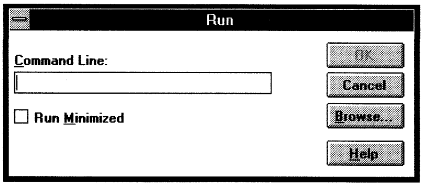
You install from diskettes using the Setup command, just as you do for other Windows applications.

❖ To install from diskettes created by the Install Diskette Builder:

1 Insert DISK1 into the diskette drive.

2 Select File➤Run from the Windows Program Manager menu bar.

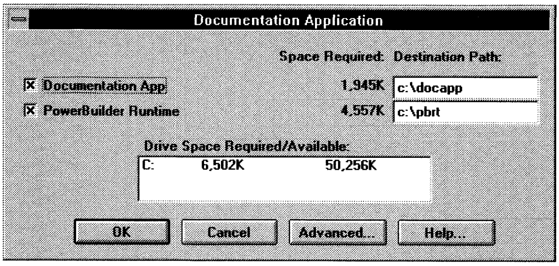
The Run dialog box displays.



3 Type ***drivename*:setup**

4 Click OK.

The Setup program initializes and displays the Documentation Application dialog box. The Setup program displays dialog boxes with the name you specified in the Install Diskette Builder's Caption box.



	Space Required:	Destination Path:
☒ Documentation App	1,945K	c:\docapp
☒ PowerBuilder Runtime	4,557K	c:\pbrt

Drive Space Required/Available:

C: 6,502K 50,256K

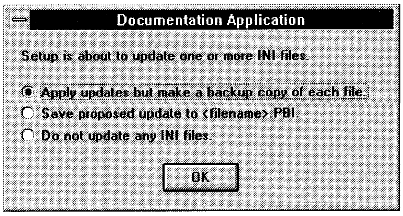
5 Select installation options, as desired. You can:

- Deselect installation components.
- Rename destination paths.
- Click Advanced to display a dialog box that allows you to control updates to the WINDOWS and WINDOWS\SYSTEM directories.
- Click Help, which displays the Read Me file, as specified in the Install Diskette Builder's Read Me File box.

6 Click OK.

The Setup program creates directories, uncompresses files, and copies them to the appropriate directory. A dialog box informs you of installation progress, prompting for additional diskettes, as needed.

7 If INI files or your AUTOEXEC.BAT file will be updated, the Setup program prompts you for how to proceed.

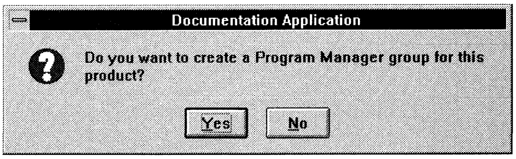


Setup is about to update one or more INI files.

● Apply updates but make a backup copy of each file.
○ Save proposed update to <filename>.PBI.
○ Do not update any INI files.

8 Select the appropriate radio button and click OK.

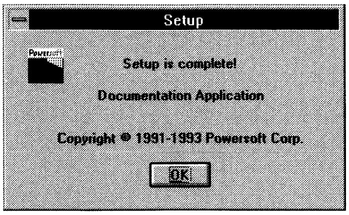
9 If the configuration file includes Program Group specifications, the Setup program asks whether to create it.



Do you want to create a Program Manager group for this product?

10 Click Yes or No, as desired.

The Setup dialog box displays when the install process completes.



Setup is complete!

Documentation Application

Copyright © 1991-1993 Powersoft Corp.

11 Click OK.

The Windows Program Manager displays

Install Diskette Builder menu bar

The Install Diskette Builder contains a menu bar. This section describes the menu items unique to the Install Diskette Builder.

For information on the Help menu, see the discussion in Chapter 2, "Cross Reference."

File menu

The File menu contains choices that perform actions on Install Diskette Builder windows.

Menu item	Activity	Shortcut key
New	Displays the New File Name dialog box, which you use to name the drive, directory, and name of a configuration (CFG) file to be created by the Install Diskette Builder	
Open	Displays the Select File dialog box, which you use to select the CFG file to be opened	
Save	Saves the current CFG file	
Exit	Closes the Install Diskette Builder	ALT+F4

Actions menu

The Actions menu contains choices that display windows and dialog boxes allowing you to perform install diskette creation activities.

Menu item	Activity
Define Reusable Component	Displays the Reusable Components window, which you use to specify components containing lists of modules included in installation diskettes for multiple applications. You typically use reusable components to specify executable modules for PowerBuilder and your DBMS.

Menu item	Activity
Define Progman Group	Displays the Program Group dialog box, which you use to specify whether the installation process will create a Program Group and to name the Program Items that display in the Program Group.
Read Project Object	Displays the Select Application dialog box, which in turn displays the Select Project Object dialog box, allowing you to select a project object whose EXE and PBD files will populate the component list.
Create Diskette Images	Displays the Create Diskette Images dialog box, which you use to control the creation of diskette images for an application.
Create Diskettes	Displays the Create Diskettes dialog box, which you use to control diskette creation.

Index