



Open Tools from Sybase, Inc.

PowerBuilder

Synchronizer

User's Guide

Version 6

Power Builder®

AB0881

October 1997

Copyright © 1991-1997 Sybase, Inc. and its subsidiaries.

All rights reserved.

Printed in Ireland.

Information in this manual may change without notice and does not represent a commitment on the part of Sybase, Inc. and its subsidiaries.

The software described in this manual is provided by Powersoft Corporation under a Powersoft License agreement. The software may be used only in accordance with the terms of the agreement.

No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc. and its subsidiaries.

Sybase, Inc. and its subsidiaries claim copyright in this program and documentation as an unpublished work, revisions of which were first licensed on the date indicated in the foregoing notice. Claim of copyright does not imply waiver of other rights of Sybase, Inc. and its subsidiaries.

ClearConnect, Column Design, ComponentPack, InfoMaker, ObjectCycle, PowerBuilder, PowerDesigner, Powersoft, S-Designor, SQL SMART, and Sybase are registered trademarks of Sybase, Inc. and its subsidiaries. Adaptive Component Architecture, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Warehouse, AppModeler, DataArchitect, DataExpress, Data Pipeline, DataWindow, dbQueue, ImpactNow, InstaHelp, Jaguar CTS, jConnect for JDBC, MetaWorks, NetImpact, Optima++, Power++, PowerAMC, PowerBuilder Foundation Class Library, Power J, PowerScript, PowerSite, Powersoft Portfolio, Powersoft Professional, PowerTips, ProcessAnalyst, Runtime Kit for Unicode, SQL Anywhere, The Model For Client/Server Solutions, The Future Is Wide Open, Translation Toolkit, UNIBOM, Unilib, Uninull, Unisep, Unistring, Viewer, WarehouseArchitect, Watcom, Watcom SQL Server, Web.PB, and Web.SQL are trademarks of Sybase, Inc. or its subsidiaries. Certified PowerBuilder Developer and CPD are service marks of Sybase, Inc. or its subsidiaries. DataWindow is a patented proprietary technology of Sybase, Inc. or its subsidiaries.

AccuFonts is a trademark of AccuWare Business Solutions Ltd.

All other trademarks are the property of their respective owners.

Contents

| | | |
|----------|--|-----------|
| | About This Book | v |
| 1 | Getting Started | 1 |
| | About the Synchronizer | 2 |
| | Basic steps of using Sync | 4 |
| | Installing Sync | 5 |
| 2 | Using Sync Builder | 7 |
| | Preparing to use Sync Builder | 8 |
| | Starting Sync Builder | 10 |
| | Building a Sync data file | 11 |
| | Creating and modifying a Sync data file | 11 |
| | Adding instructions to the Sync data file | 13 |
| | Testing a Sync data file | 21 |
| 3 | Running the Sync Runtime Executable | 23 |
| | Why run the Sync runtime executable | 24 |
| | Running the Sync runtime executable | 25 |
| | What's next | 28 |
| 4 | Using the Sync ActiveX | 29 |
| | About the Sync ActiveX | 30 |
| | Sync ActiveX properties and methods | 31 |
| | Using the Sync ActiveX in a Web page | 32 |
| | Inserting the Sync ActiveX in HTML | 32 |
| | Changing security settings in your browser | 37 |
| | Using the Sync ActiveX in a window | 38 |
| | What's next | 41 |
| 5 | Deploying Synchronizing Applications and HTML Pages | 43 |
| | About deploying applications and Web pages that synchronize | 44 |
| | Deploying an application that synchronizes its files | 45 |

Deploying an HTML page containing the Sync ActiveX 47
Deploying an application with the Sync ActiveX in a window..... 48
Additional files that may be needed on users' systems 49

6 **Sync Data Files** **51**
 About Sync data files 52
 Sync data file structure..... 53
 [Sync] section 54
 [Log] section 56
 [Start] section 57
 [Variables] section 58
 [End] section 59

About This Book

Subject

This book describes the Synchronizer and how to use it for synchronizing applications and synchronizing on a Web page.

Audience

This book assumes you are a PowerBuilder developer creating and deploying Windows applications or a system administrator working with a PowerBuilder developer to deploy applications in a corporate environment.

Getting Started

About this chapter

This chapter describes the Synchronizer, the basic steps for using it, and installation information.

Contents

| Topic | Page |
|---------------------------|-------------|
| About the Synchronizer | 2 |
| Basic steps of using Sync | 4 |
| Installing Sync | 5 |

About the Synchronizer

The Synchronizer (Sync) is software that synchronizes two sets of files. Sync compares source files to destination files and then copies the latest source files to their destinations so that the files match.

Why use Sync

For synchronizing deployed applications Sync's primary purpose is to synchronize files for your deployed PowerBuilder applications. Without Sync, it's difficult to maintain and deploy applications. With Sync, you can update your users' application files to match the latest application files—so you don't have to do it.

When a user starts a synchronizing application, Sync runs first to synchronize application files, and then Sync runs the application. So every time a PowerBuilder application runs, the application will run with the latest version of all the files needed for the application.

For example, before a user's application starts, Sync could synchronize the application's PBDs and DLLs and the PowerBuilder deployment DLLs.

For synchronizing files from a Web page You can use Sync to add synchronization ability to a Web page so that when users access the page, synchronization occurs.

For example, using the Sync ActiveX on an HTML page, Sync could synchronize the PowerBuilder window ActiveX and the PowerBuilder virtual machine (PBVM60.DLL) before displaying a child window in the PowerBuilder window ActiveX.

What Sync does

Sync does its work in five basic steps, based on your instructions:

- 1 Looks for synchronization instructions (usually found in a Sync data file) that specify the location of the source files and the destination files.
- 2 (Optional) Displays a status window and logs the execution of instructions.
- 3 Compares the source files to the destination files by checking date/time stamps, sizes, and version information.
- 4 Copies changed source files (including source files that don't exist on the destination) to the destination.
- 5 (Optional) Runs an application when synchronization completes.

Sync components

Sync has three components:

- ◆ **Sync Builder** You use this to create and test Sync data files. With Sync Builder, you don't have to worry about the structure of a Sync data file. You concentrate on what you want the Sync data file to do.

Sync Builder runs on 32-bit Windows only.

- ◆ **Sync runtime executable** You use this when you deploy an application that synchronizes.

The Sync runtime executable runs on 16-bit and 32-bit Windows.

- ◆ **Sync ActiveX** You use this to provide synchronization in an application window or in an HTML page.

The Sync ActiveX runs on 32-bit Windows only.

Not for installations

Sync is not designed for creating installations. Use an installation building tool for that.

Although Sync could be used to install some applications, installing applications usually involves more than copying files. Sync can't perform standard installation activities such as registry updates, INI file updates, and for some 16-bit applications, updating AUTOEXEC.BAT.

Basic steps of using Sync

Using Sync involves these procedures:

| Goal | What you do | What you use | See |
|--|---|-------------------------|--|
| Create Sync data files | Set up synchronization instructions in a Sync data file and test the instructions | Sync Builder | Chapter 2, "Using Sync Builder" |
| Provide synchronization ability for deployed applications | Test deployment | Sync runtime executable | Chapter 3, "Running the Sync Runtime Executable" |
| | Deploy the synchronizing application | Sync runtime executable | Chapter 5, "Deploying Synchronizing Applications and HTML Pages" |
| Add synchronization ability to a Web page or an application window | Add the Sync ActiveX to a Web page or a PowerBuilder window | Sync ActiveX | Chapter 4, "Using the Sync ActiveX" |
| | Deploy the HTML page or the PowerBuilder application | Sync ActiveX | Chapter 5, "Deploying Synchronizing Applications and HTML Pages" |

Installing Sync

If you choose to install Sync when you install PowerBuilder, it is installed by default in the PowerBuilder Sync directory.

Wininet files

Sync requires Wininet files on your system. When Sync is installed, the following Wininet files are also installed in your Windows 95 or Windows NT system directory:

HLINK.DLL
HLINKPRX.DLL
INLOADER.DLL
OLEAUT32.DLL
STDOLE2.TLB
URLMON.DLL
WININET.DLL

MFC Runtime modules

When you install Sync, the Sync ActiveX is:

- ◆ Installed in your Windows 95 or Windows NT system directory
- ◆ Registered in the registry

The Sync ActiveX requires the Microsoft Foundation Class (MFC) Runtime modules. These modules are usually found in the Windows 95 or Windows NT system directory.

When the Sync ActiveX is installed, the following MFC DLLs are updated on your system if they are not present or you have earlier versions:

MFC42.DLL
MFVCIRT.DLL
MSVCRT.DLL
OLEPRO32.DLL

Using Sync Builder

About this chapter

This chapter tells you how to use Sync Builder to build and test Sync data files.

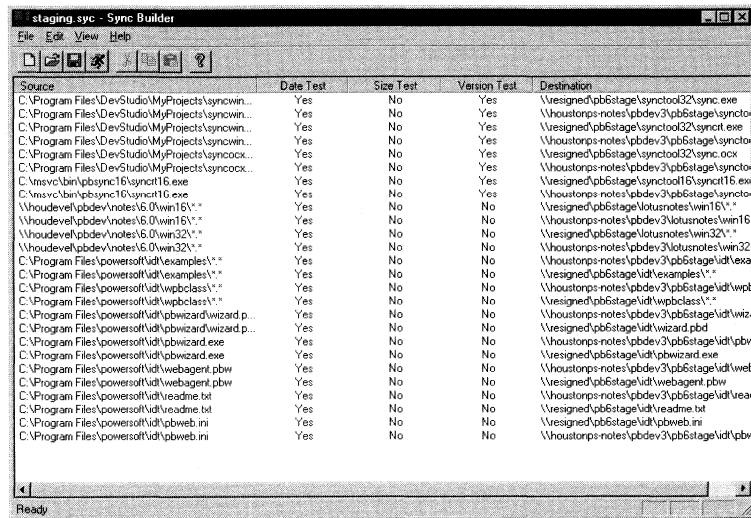
Contents

| Topic | Page |
|-------------------------------|-------------|
| Preparing to use Sync Builder | 8 |
| Starting Sync Builder | 10 |
| Building a Sync data file | 11 |
| Testing a Sync data file | 21 |

Preparing to use Sync Builder

You use Sync Builder (SYNC.EXE) to create and test Sync data files. The Sync data files you create can run in SyncBuilder, but they will be used later by the Sync runtime executable and the Sync ActiveX.

As you work in Sync Builder, you'll see information in the workspace about the source files you want Sync to copy and the destinations you are copying them to:



Deciding whether to use short or long filenames

If you will be using the 16-bit Sync runtime executable to run Sync data files, you must create Sync data files that reference directories with short filenames. You can do this by browsing in Sync Builder to directories that have short filenames or by typing a short filename instead of browsing.

When you use Sync Builder to create Sync data files, you may always want to reference directories with short filenames. Then your Sync data files can be used by both the 16-bit and the 32-bit Sync runtime executable.

Deciding about source files and destinations

When you use Sync Builder, the most basic instructions are those that tell Sync what source files to copy and where to copy them. So you need to decide where to put source files and what their destinations will be.

Source files Sync Builder, the Sync runtime executable, and the Sync ActiveX can all copy files from the same sources. So when you are creating a Sync data file in SyncBuilder, you don't have to be concerned about what Sync component will be running the Sync data file.

The sources are:

| Source | Example |
|--|--|
| FTP server | FTP://FTP.POWERSOFT.COM |
| Local drive | C:\FOLDER\FILENAME |
| Network drive (using the path or a mapped drive letter) | \\SERVER\DRIVE\FOLDER\FILENAME or \\SERVER\SHARENAME\FOLDER\FILENAME or I:\FOLDER\FILENAME |

Using paths or mapped drive letters

You can specify network drive paths or a mapped drive letter. Using paths is better because mapped drive letters can change.

When connecting to an FTP server, Sync attempts to connect using the user ID *anonymous*. If the FTP server requires a password, Sync provides the user's e-mail address, which is a common Internet convention. Sync looks for the user's e-mail address in this location:

```
\\MyComputer\HKey_Current_User\Software\Microsoft
\Windows\CurrentVersion\Internet Settings\EmailName
```

Destinations Sync Builder, the Sync runtime executable, and the Sync ActiveX can copy source files to a local drive or a network server.

Deciding where to put Sync data files

When you create a Sync data file in Sync Builder, you can save it to a local or network drive. But eventually, the Sync data file you create will be run by the Sync runtime executable or the Sync ActiveX.

Sync components can run a Sync data file from the following locations:

| Component | Location of the Sync data file it runs |
|-------------------------|--|
| Sync Builder | Local drive Network server |
| Sync Runtime Executable | Local drive Network server |
| Sync ActiveX | Local drive Network server HTTP server |

Starting Sync Builder

If you choose to install Sync when you install PowerBuilder, a Sync button will display on the PowerBar.

If you did not install Sync

For how to add Sync to your PowerBuilder installation, see the *PowerBuilder Installation Guide*.

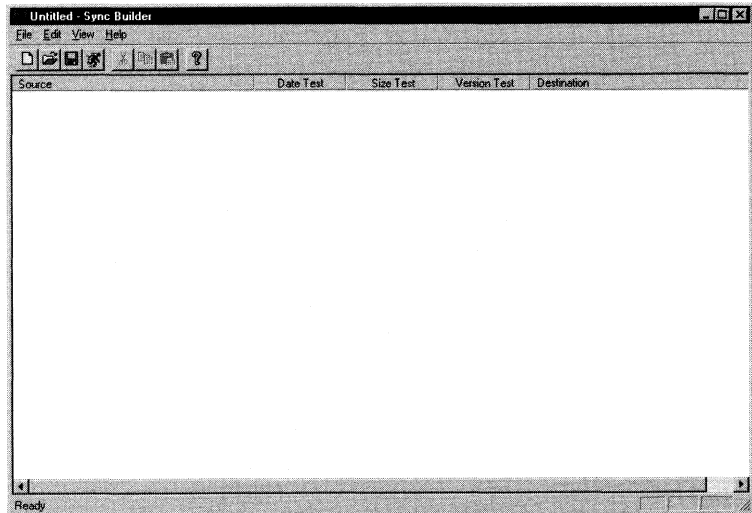
❖ To start Sync Builder



- ◆ In PowerBuilder, click the Sync button on the PowerBar. (The Sync button is also in the dropdown PowerPanel at the leftmost end of the PowerBar).

or

Select Synchronizer in the PowerBuilder programs group.



Building a Sync data file

When you build a Sync data file, you specify instructions for Sync to execute.

Creating and modifying a Sync data file

Creating a Sync data file

The following procedure gives you the basic steps for creating a Sync data file. You'll find the details about adding instructions in "Adding instructions to the Sync data file" on page 13.

❖ **To create a Sync data file:**

- 1 If the Sync Builder workspace displays instructions, select File>New to clear the workspace.
- 2 Add an instruction to copy files by displaying the popup menu and then selecting New>Sync Command.
or
Add an instruction to include a nested Sync data file by displaying the popup menu and then selecting New>Nested Sync.
- 3 (Optional) Add additional instructions to do any of the following: log the synchronization process; display a status window with or without a Cancel button; start an application when the synchronization completes; and define runtime variables.

Displaying a Status window and running an application

Although displaying a Status window and running an application are optional, you will usually want to have Sync take these actions. Displaying a Status window is a visual indicator for a user that Sync is running. And when you deploy an application that synchronizes, you want Sync to start the application after synchronization completes.

- 4 Save the Sync data file.

Modifying an existing Sync data file

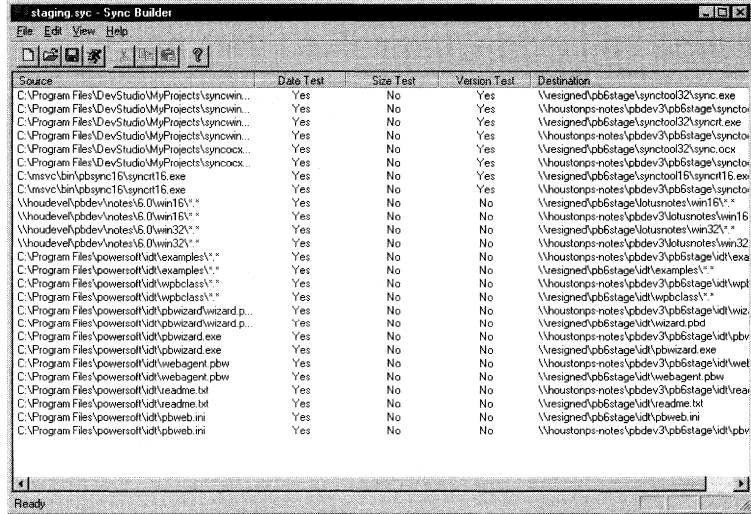
You can modify an existing Sync data file in Sync Builder.

❖ **To open an existing Sync data file:**

- 1 Select File>Open.

- In the Open dialog box, navigate to the file and select it, and then click Open.

Sync Builder opens the Sync data file. You see information in the workspace about the source files and their destinations:



- ❖ **To edit an instruction that displays in the workspace:**
 - Select the source for the instruction at the left end of the workspace and then select Properties from the popup menu.
or
Double-click the source.
 - Edit the instruction in the Command Properties dialog box.
- ❖ **To edit other instructions:**
 - Select View>Options from the menu bar.
 - Edit the instructions in the Options dialog box.
- ❖ **To cut, copy, or delete an instruction:**
 - Select the source for the instruction at the left end of the workspace and display the popup menu.
 - Select Cut, Copy, or Delete from the popup menu.
- ❖ **To paste an instruction:**
 - Point at the workspace and display the popup menu.

- 2 Select Paste from the popup menu.

Adding instructions to the Sync data file

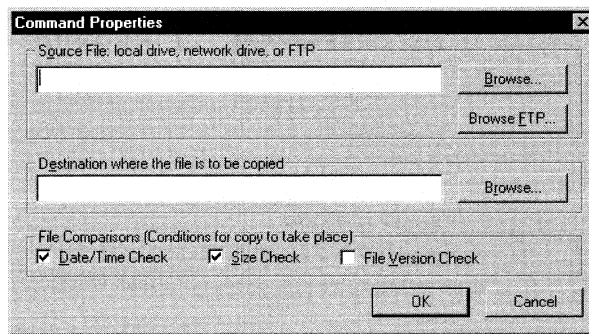
Specifying files for Sync to copy

The most basic instruction you'll specify in Sync Builder is an instruction to copy files from a source to a destination.

❖ To add an instruction to copy files:

- 1 In the workspace, display the popup menu and select New>Sync Command.

The Command Properties dialog box displays:



- 2 Type the source file specification.

or

Click the Browse button and navigate to the source.

or

Click the Browse FTP button to browse to an FTP address if your source files are on an FTP site.

If you are using paths for network drives in the source file specification, you type them. You can use wildcards. You can also use a system-defined runtime variable (`%systemdir%` or `%windowsdir%`) or a runtime variable you have defined (enclosed by `%` symbols) to represent the source.

- 3 Type the destination specification (which cannot be an FTP site).

or

Click the Browse button to find the destination.

You can use wildcards in the destination specification. You can also use a system-defined runtime variable (`%systemdir%` or `%windir%`) or a runtime variable you have defined (enclosed by `%` symbols) to represent the destination.

Here are valid ways to specify copying files from a source to a destination:

| Source | Destination |
|----------------------|--------------------|
| I:\SOURCE\MYFILE.TXT | C:\DEST\MYFILE.TXT |
| I:\SOURCE\MYFILE.TXT | C:\DEST\ |
| I:\SOURCE\MYFILE.TXT | C:\DEST*.* |
| I:\SOURCE*.* | C:\DEST\ |
| I:\SOURCE*.* | C:\DEST*.* |

Invalid method

Specifying a destination as `C:\TEMP` (without the trailing backslash) is not valid, because it's not clear whether the destination is a folder or a file.

- 4 Specify how you want Sync to compare source and destination files by selecting or clearing the File Comparisons checkboxes: Date Time Check, Size Check, and File Version Check.

For FTP sources

Date Time checking is against the date only; File Version checking does not apply.

- 5 Click OK.

Nesting Sync data files

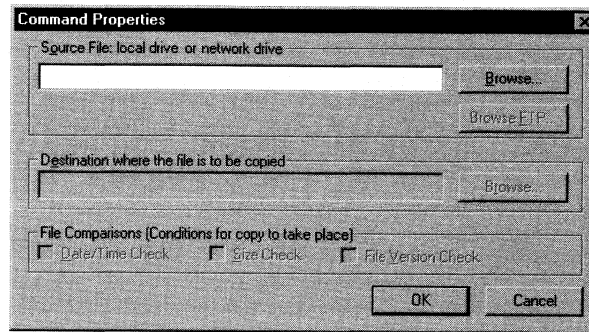
You can nest another Sync data file in the current file you are creating. Nesting Sync data files allows you to set up Sync data files for a particular application or for a set of common files (such as the PowerBuilder deployment DLLs) and then nest the Sync data files in other Sync data files (such as a departmental Sync data file).

Runtime variables in nested Sync data files

A nested Sync data file can have its own runtime variables that were defined when the file was created. A nested Sync data file can reference its own runtime variables and also runtime variables defined in the parent Sync data file. It's better not to duplicate runtime variable names in the nested and parent Sync data files. If you do, the definition in the nested file has priority.

❖ To add an instruction to nest a Sync data file:

- 1 In the workspace, display the popup menu and select New>Nested Sync:



- 2 Type the specification for the Sync data file you want to nest.
or
Click the Browse button and find the Sync data file.

You can use a system-defined runtime variable (%systemdir% or %windowsdir%) or a runtime variable you have defined (enclosed by % symbols) in the source.

Nested Sync data files can't be on FTP servers

The Sync data file you want to nest must be on a local or network drive. It can't be on an FTP server. But the nested Sync data file can include instructions for copying files from FTP servers.

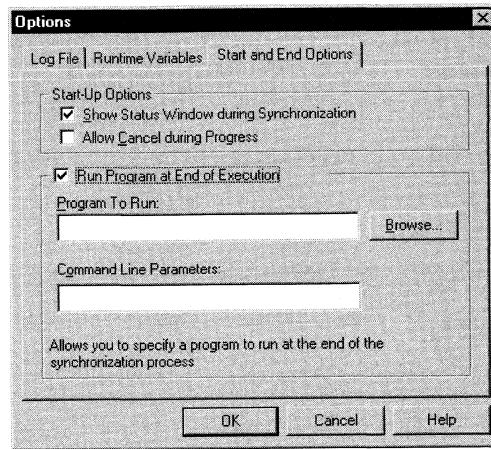
- 3 Click OK.

Running an application when synchronization completes

When the Sync data file you are creating completes the synchronization process, you will usually want to run an application. You can specify command-line parameters for running the application if the application has them.

❖ **To add an instruction to run an application:**

- 1 Select View>Options from the menu bar and then select the Start and End Options tab.
- 2 Click the Run Program at End of Execution checkbox:



- 3 Type the file specification for the application you want to run.
or
Click the Browse button to find the application's executable file.

You can use a system-defined runtime variable (%systemdir% or %windowsdir%) or a runtime variable you have defined (enclosed by % symbols) to represent the file to execute.
- 4 If the program you are running has command-line parameters and you want to use one or more of these, type the specification (separating each parameter from the next with a space).

For Microsoft Word, for example, a command-line parameter is the location and name of a Word document.

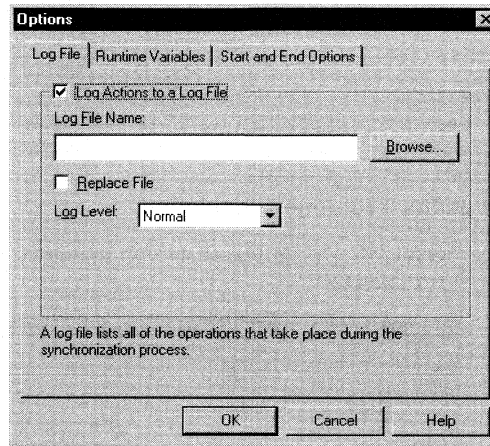
You can use a system-defined runtime variable (%systemdir% or %windowsdir%) or a runtime variable that you have defined (enclosed by % symbols) to represent command-line parameters.
- 5 Click OK.

Logging the synchronization

When the Sync data file you are creating runs, you may want it to log what Sync is executing.

❖ To add an instruction for logging the synchronization:

- 1 Select View>Options from the menu bar and then select the Log File tab.
- 2 Click the Log Actions to a Log File checkbox:



- 3 Type the file specification for an existing log file or a log file you want Sync to create.

or

Click the Browse button to specify the destination for an existing log file or a log file you want Sync to create.

You can use a system-defined runtime variable (%systemdir% or %windowsdir%) or a runtime variable that you have defined (enclosed by % symbols) in the log file specification.

- 4 To replace the log file whenever the Sync data file executes, select the Replace File checkbox. To append to the log file, clear the checkbox.
- 5 In the Log Level dropdown listbox, select a log level: Errors Only, Normal, or Verbose.
- 6 Click OK.

Defining runtime variables

You can define runtime variables to represent a source, destination, log file, application to run, or command-line parameters associated with the application. It's helpful to use a runtime variable for file specifications or command-line parameters that are long.

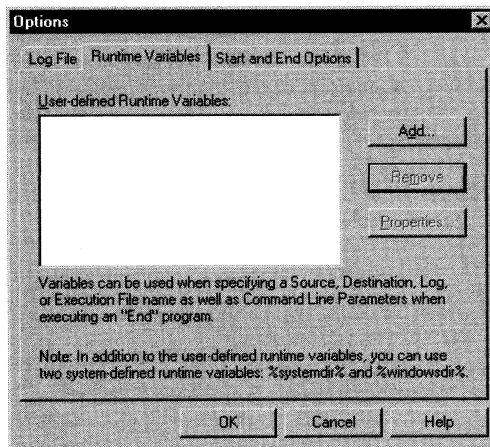
System-defined runtime variables

You can also use two system-defined runtime variables. The `%systemdir%` runtime variable represents the user's Windows system directory. The `%windowsdir%` runtime variable represents the user's Windows directory.

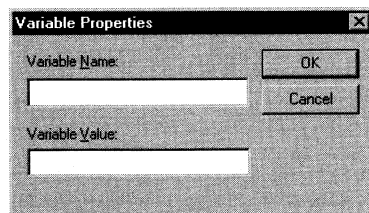
System-defined runtime variables are resolved at runtime to whatever value is appropriate for the user's system. For example, `%systemdir%` resolves to the user-specific Windows system directory. The value depends on whether the user is running Windows 95 or Windows NT.

❖ To define a runtime variable:

- 1 Select View>Options from the menu bar and then select the Runtime Variables tab:



- 2 Click the Add button.



- 3 Type a variable name and a variable value (you can't use a % symbol in the name) that specifies a source, destination, log file, program to run, or command-line parameter for the program, then click OK.

The variable definition displays in the User-Defined Runtime Variables box in the Options dialog box.

- 4 Click OK.

❖ **To use a runtime variable you've defined:**

- ◆ In the Options dialog box, when you specify a source, destination, log file, program to run, or command-line parameter for the program, instead of typing specifications or browsing for them you type a runtime variable name enclosed by % symbols:

```
%RuntimeVariableName%
```

❖ **To delete a runtime variable:**

- 1 Select the definition in the Options dialog box.
- 2 Click Remove.

❖ **To edit a runtime variable:**

- 1 Select the definition in the Options dialog box.
- 2 Click Properties.
- 3 Modify the variable name or value and click OK.

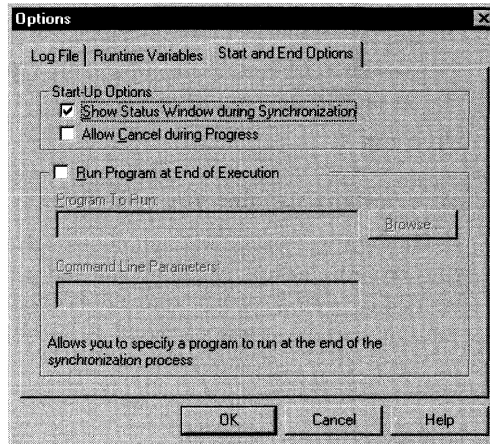
Displaying a status window with or without a Cancel button

You can instruct Sync to display a status window (with or without a Cancel button). Most of the time you should display a status window so users will have a visual indication that synchronization is happening. If you also specify logging, users will see the logging information displaying in the status window.

❖ **To add an instruction to display a status window with or without a Cancel button:**

- 1 Select View>Options from the menu bar and then select the Start and End Options tab.

- 2 Click the Show Status Window During Synchronization checkbox:



- 3 If you want to allow the user to click a Cancel button to cancel the synchronization process before it completes, click the Allow Cancel During Progress checkbox.
- 4 Click OK.

Testing a Sync data file

You should test Sync data files before you use them to be sure they are copying the appropriate source files to the destinations you want.

To test a Sync data file, you run it.

❖ **To run a Sync data file:**

- 1 Open a Sync data file.
- 2 Click the Run button.

After you run the file, you should verify that Sync has done what you intended. If not, modify the instructions.

CHAPTER 3

Running the Sync Runtime Executable

About this chapter

This chapter tells you how to run the Sync runtime executable to perform synchronization.

Contents

| Topic | Page |
|-------------------------------------|-------------|
| Why run the Sync runtime executable | 24 |
| Running the Sync runtime executable | 25 |
| What's next | 28 |

Why run the Sync runtime executable

When you run the Sync runtime executable, it performs the synchronization specified in a Sync data file.

When you deploy a synchronizing application, the Sync runtime executable and the Sync data file will most often reside on your corporate network.

Running the Sync runtime executable

You can run the Sync runtime executable any of three ways:

| Ways to run the Sync runtime executable | You'll do it this way |
|---|-----------------------|
| Passing to the Sync runtime executable a reference to a Sync data file | Most of the time |
| Associating SYC files with the Sync runtime executable | Sometimes |
| Passing to the Sync runtime executable a command-line argument with one synchronization instruction | Very infrequently |

In the examples in this section, SYNCRT.EXE is used for the Sync runtime executable; for 16-bit systems, you should replace SYNCRT.EXE with SYNCRT16.EXE.

Passing a reference to a Sync data file

You can run Sync from the command line and give Sync more than one instruction by passing a reference to a Sync data file containing the instructions you want Sync to execute.

The command format is:

```
SYNCRT.EXE SyncDataFileSpec { /B /LOG LogFileSpec }
```

SyncDataFileSpec and *LogFileSpec* are the location and name of the Sync data file and log file. When you specify these, you can use the path of the network drive:

```
\\SERVER\DRIVE\FOLDER\FILENAME
```

or the letter the network drive is mapped to:

```
I:\FOLDER\FILENAME
```

Using the path is better, because the mapped drive can change.

After you pass the reference to the Sync data file, you can pass either or both of the following optional parameters:

| Parameter | What it makes Sync do |
|-------------------------|---|
| /B | Run in batch mode without displaying a status window |
| /LOG <i>LogFileSpec</i> | Log the synchronization process in the specified log file |

The optional parameters will override instructions in the Sync data file to display a status window or omit logging.

❖ **To pass a reference to a Sync data file:**

- ◆ At the command line, enter a command that runs SYNCRT.EXE and references a Sync data file.

The command line can be in DOS, in the Windows Start>Run menu, or in a shortcut's properties.

For example, to execute the Sync data file MYSYNC.SYC:

```
CAFE\APP\SYNC\SYNCRT.EXE
CAFE\APP\SYNCDATA\MYSYNC.SYC /B
```

Associating SYC files with the Sync runtime executable

You can associate the SYC extension for Sync data files with the Sync runtime executable.

❖ **To associate SYC files with the Sync runtime executable:**

- 1 Use your operating system's method to associate SYC files with SYNCRT.EXE or SYNCRT16.EXE.

For example, in Windows 95 you can associate an SYC file with SYNCRT.EXE by opening Explorer, selecting View>Options from the menu bar and selecting the File Types tab, selecting SYC File from the list of registered file types, and then editing the application used to perform the Open action.

- 2 Double-click an SYC file or a shortcut associated with an SYC file to run the Sync runtime executable.

Passing a command-line argument

You can run Sync from the command line by passing a command-line argument with one synchronization instruction to the Sync runtime executable. The instruction specifies the source files, the destination, and optional parameters.

The command format is:

```
SYNCRT.EXE /SRC SourceFiles /DEST DestinationFiles { Parameters }
```

SourceFiles and *DestinationFiles* are the location and name of the sources files and the destination files respectively. For example, *SourceFiles* could be one file:

```
I:\SOURCE\PBVM60.DLL
```

or many files:

```
I:\SOURCE\*.DLL
```

You can use the path of a network drive:

```
\\SYNC\SOURCE\PB\PBVM60.DLL
```


or a mapped network drive:

```
I : \PB\PBVM60.DLL
```

After you specify the destination files on the command line, you can pass any or all of the following optional parameters:

| This parameter | Tells Sync to |
|----------------------|---|
| /B | Run in batch mode without displaying a status window |
| /D | Compare source and destination files by checking the dates |
| /S | Compare source and destination files by checking the sizes |
| /V | Compare source and destination files by checking the version information embedded in DLL or EXE files |
| /LOG <i>FileName</i> | Log the synchronization process in the specified log file |

❖ **To pass a command-line argument:**

- ◆ At the command line, enter a command that runs SYNCRT.EXE and includes one source specification and one destination specification.

The command line can be in DOS, in the Windows Start>Run menu, or in a shortcut's properties.

For example, to update the PB60.EXE file:

```
C : \SYNCRT.EXE /SRC
CAFE\APP\PB\SOURCE\PB60.EXE
/DEST C:\PROGRAM FILES\POWERSOFT\PB6\PB60.EXE
/D/S/LOG MYLOG.LOG
```

To update many DLLs, you can use a wildcard:

```
C : \SYNCRT.EXE /SRC CAFE\APP\PB\SOURCE\*.DLL
/DEST C:\PROGRAM FILES\
```

What's next

For information about how to deploy an application that synchronizes application files before it starts, see Chapter 5, "Deploying Synchronizing Applications and HTML Pages".

About this chapter

This chapter explains how to use the Sync ActiveX control to provide synchronization in an application window or on a Web page.

Contents

| Topic | Page |
|--------------------------------------|-------------|
| About the Sync ActiveX | 30 |
| Sync ActiveX properties and methods | 31 |
| Using the Sync ActiveX in a Web page | 32 |
| Using the Sync ActiveX in a window | 38 |
| What's next | 41 |

About the Sync ActiveX

The Sync ActiveX (SYNC.OCX) is a 32-bit nonvisual component that you use to provide synchronization on a Web page or in a window of a 32-bit application.

Inserting the Sync ActiveX

You can insert the Sync ActiveX in:

- ◆ An HTML page
- ◆ A PowerBuilder window (or a window built with another application development tool that supports ActiveX)

When you insert the Sync ActiveX in an application window

Because a running application's files are locked and cannot be updated, you cannot use the Sync ActiveX in an application window to synchronize that application's files.

Runs a Sync data file

The Sync ActiveX runs a Sync data file the same way that Sync Builder (SYNC.EXE) and the Sync runtime executable (SYNCRT.EXE and SYNCRT16.EXE) do. You create the Sync data file in Sync Builder.

FOR INFO For information about using Sync Builder to create a Sync data file, see Chapter 2, "Using Sync Builder".

The Sync ActiveX can run a Sync data file in any of three locations:

- ◆ Local drive
- ◆ Network drive
- ◆ HTTP server

The Sync ActiveX cannot run a Sync data file located on an FTP server.

When you insert the Sync ActiveX in an HTML page, you specify the Sync data file as a property of the Sync ActiveX.

Sync ActiveX properties and methods

When you use the Sync ActiveX in a window or on a Web page, you'll be using the properties and methods of the Sync ActiveX.

Properties

These are the Sync ActiveX properties:

| Sync ActiveX property | Description |
|------------------------------|--|
| ShowStatus | Boolean specifying whether a status window displays when the Sync ActiveX runs a Sync data file |
| SyncFileName | String specifying the name and location of the Sync data file (a local or network drive or an HTTP server) |

Methods

These are the Sync ActiveX methods:

| Sync ActiveX method | Description |
|----------------------------|--|
| AboutBox() | Displays an About box for the Sync ActiveX |
| Execute() | Performs the instructions in the specified Sync data file (specified by the SyncFileName property) |

Using the Sync ActiveX in a Web page

If you want to perform synchronization on a Web page, you insert the Sync ActiveX in an HTML page and display the page in a Web browser that supports ActiveX.

For example, on a Web page you could synchronize the PowerBuilder virtual machine used by the PowerBuilder window plug-in. You insert the Sync ActiveX in the HTML page using the Object statement, specify the properties, and code the Execute statement. When a user opens the HTML page in a browser that supports ActiveX, the Sync ActiveX executes a Sync data file that updates the PowerBuilder virtual machine before displaying the HTML page.

Inserting the Sync ActiveX in HTML

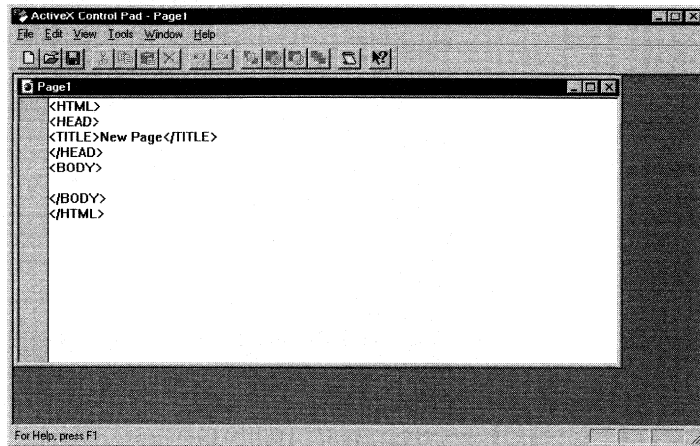
The following instructions show you how to insert the Sync ActiveX in an HTML page using the ActiveX Control Pad.

Other ways to code HTML

You can use another ActiveX-friendly editor such as Microsoft Front Page. An ActiveX-friendly editor does most of the HTML coding for you automatically. Whatever ActiveX-friendly editor you use, the process of inserting the Sync ActiveX in HTML is similar. You can even use a simple text editor, but you'll need expert knowledge of HTML and class information for the Sync ActiveX control.

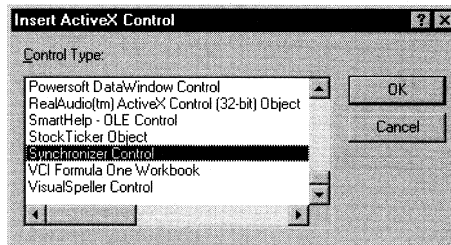
❖ **To insert the Sync ActiveX in a Web page using the ActiveX Control Pad:**

- 1 Open the ActiveX Control Pad and put the pointer before the <BODY> tag:



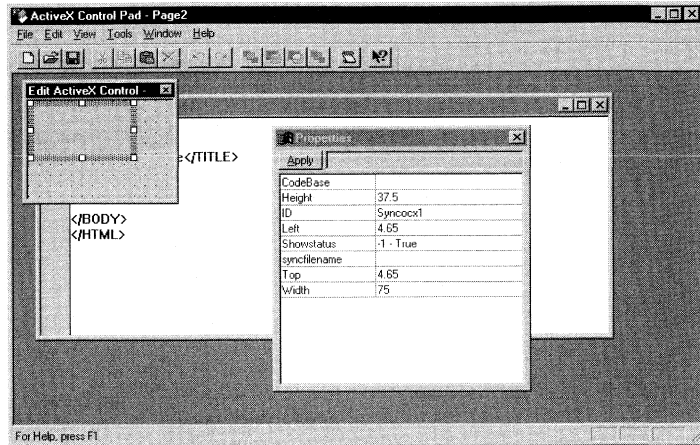
This selects the insertion point for the Sync ActiveX.

- 2 Select Edit>Insert ActiveX Control from the menu bar.
The Insert ActiveX Control dialog box displays.
- 3 Scroll to Synchronizer Control and select it:



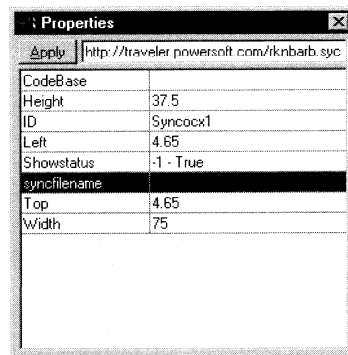
- 4 Click OK.

The Edit ActiveX Control window and the Properties window display:



- 5 Select the SyncFileName property in the Properties window, and in the box at the top of the window type the location and name of the Sync data file you want the Sync ActiveX to run. The location can be a local drive, network server, or HTTP server.

For example, in this Properties window the Sync data file has an HTTP address:



6 Click Apply.

The location and name of the Sync data file display in the properties list.

Notice the ID property Syncocx1. You'll use this ID in step 8 to code the Execute method in HTML for the Sync ActiveX.

Since the Sync ActiveX is a nonvisual object, you should change the width and height values to zero (so the Sync ActiveX doesn't take up space on the Web page).

Also notice the Show Status property (which is TRUE by default). Most of the time you'll want the value to be TRUE so the status window will display as a visual indicator that synchronization is occurring.

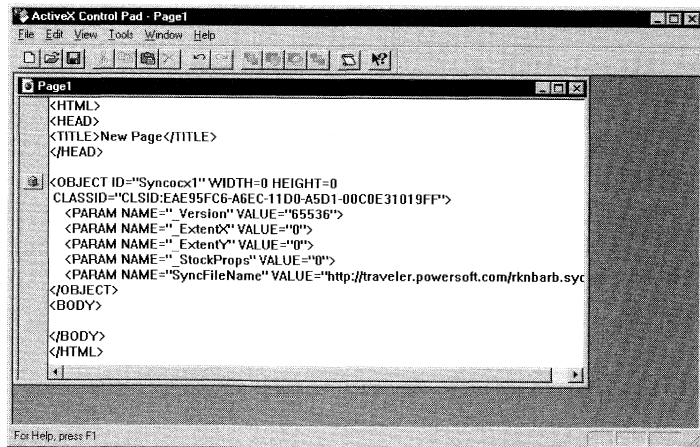
Specifying the CodeBase property

If you want the Sync ActiveX to download and register automatically when a user displays your HTML page, you must specify the CodeBase property.

FOR INFO For how to do this, see your Web browser documentation.

7 Close the Edit ActiveX Control window.

The code that's needed to place the Sync ActiveX on the HTML page displays:



```

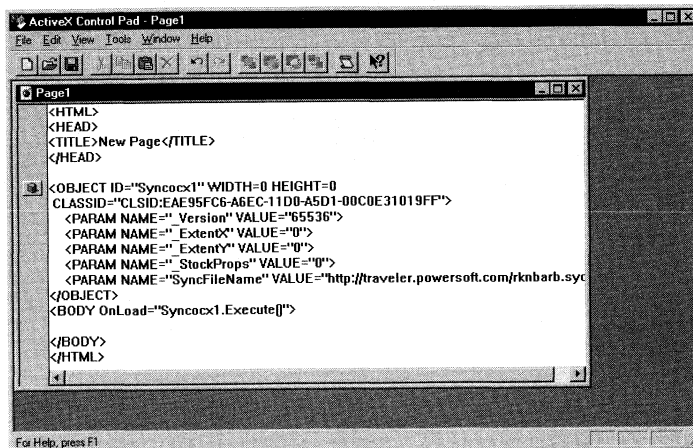
ActiveX Control Pad - Page1
File Edit View Tools Window Help
Page1
<HTML>
<HEAD>
<TITLE>New Page</TITLE>
</HEAD>
<OBJECT ID="Syncocx1" WIDTH=0 HEIGHT=0
CLASSID="{CLSID:EAE95FC6-A6EC-11D0-A5D1-00C0E31019FF}"
<PARAM NAME=".Version" VALUE="65536">
<PARAM NAME=".ExtentX" VALUE="0">
<PARAM NAME=".ExtentY" VALUE="0">
<PARAM NAME=".StockProps" VALUE="0">
<PARAM NAME="SyncFileName" VALUE="http://traveler.powersoft.com/rknbarb.sync">
</OBJECT>
<BODY>
</BODY>
</HTML>
For Help, press F1

```

The Sync ActiveX is now on the page.

8 Inside the HTML <BODY> tag, type the JavaScript OnLoad event handler to execute the Sync ActiveX:

```
OnLoad=" Syncocx1.Execute() "
```



This executes the Sync ActiveX.

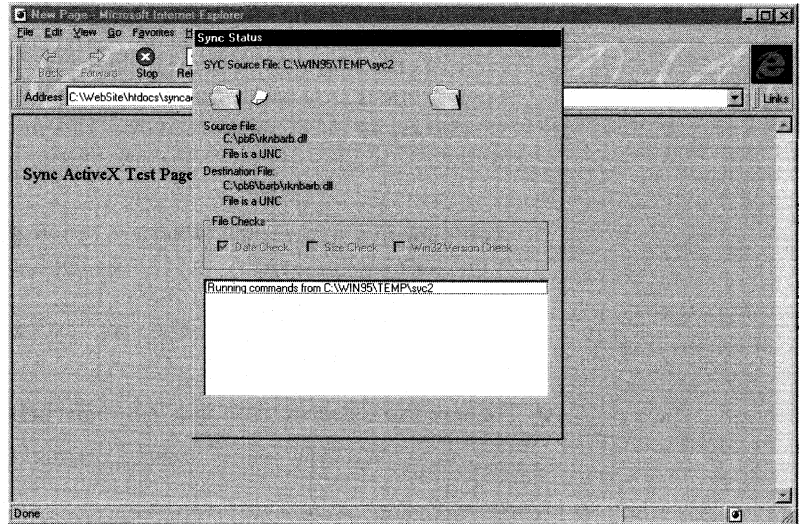
- 9 Save the HTML page in the directory that your Web server uses for HTML documents.

For example, if you are using O'Reilly's WebSite as your HTML server, the directory is WEBSITE\HTDOCS.

❖ **To test the Web page in your Web browser:**

- 1 Start your Web server.

- 2 In your browser, open the HTM file that has the Sync ActiveX in it.
The Sync ActiveX runs and executes the instructions in the Sync data file you specified:



FOR INFO If the Sync ActiveX won't load and run in your browser, see "Changing security settings in your browser" next.

Changing security settings in your browser

Lowering security levels

If the Sync ActiveX won't load and run in a browser, users may need to lower some browser security settings.

❖ **To allow ActiveXs to write to a hard disk:**

- 1 In the browser, find the dialog box where security options are set.
- 2 Reduce the security level to medium or none.

Enabling ActiveXs

Users may be unable to load the Sync ActiveX if the browser has ActiveXs disabled.

❖ **To enable ActiveXs:**

- 1 In the browser, find the dialog box where security options are set.
- 2 Modify ActiveX options to enable ActiveXs, allow downloading, and run scripts.

Using the Sync ActiveX in a window

You may want to have an application perform synchronization within the application. To do this, you insert the Sync ActiveX in a PowerBuilder window or in a window built with another application development tool.

For example, you may be using a laptop computer that is running a PowerBuilder application that accesses a SQL Anywhere database. You may want to set up the application so that the Sync ActiveX can update the SQL Anywhere runtime files with the latest files each time you start the application.

You cannot synchronize files for a running application

You can't use the Sync ActiveX in an application for the purpose of updating the application files themselves. The files you would need to update will be locked because the application is running.

❖ To use the Sync ActiveX in a PowerBuilder window:

1 In the Window painter, select Controls>OLE from the PowerBar.
or
Click the OLE button in the PainterBar.

2 In the Insert Object dialog box on the Insert Control tab, select the Synchronizer Control and click OK.
You return to the Window painter.

3 Click the window to insert the Sync ActiveX.
A visual representation of the nonvisual Sync ActiveX displays in the window.

Depending on the application you are developing, at this point you can specify the Sync ActiveX properties or you can provide for their specification programmatically so that a user can specify the properties. For example, you can allow the user to specify the Sync data file in a dialog box and then click an Execute button to run the Sync ActiveX.

FOR INFO The rest of these steps assume that you are specifying the properties of the Sync ActiveX. For information about how you can provide for their specification programmatically, see the examples that follow this procedure.

4 Right-click the control and select OCX Properties from the popup menu.

The Control Properties dialog box displays.

5 Specify the properties of the Sync ActiveX:

- ◆ Type the name and location of the Sync data file to specify the SyncFileName property.
 - ◆ Select or clear the Show Status Dialog checkbox to specify the ShowStatus property.
- 6 Click OK to close the Control Properties dialog box.
 - 7 In the window, select Script from the popup menu and then write the code to call the Sync ActiveX Execute function in the window's Open event:

```
ole_1.object.Execute()
```

You can type the code directly or you can click the Browse Object button on the PainterBar to access the PowerBuilder Browser so that you can paste functions. You can see the Sync ActiveX class information, properties, and functions in the Browser by clicking the OLE tab and then expanding the OLE Custom Controls list and then the Synchronizer Control list.

Examples: adding controls to a window to allow the user to specify properties

The following examples show the scripts associated with controls you can add to a window in which you have inserted the Sync ActiveX. These controls allow the user to specify the Sync data file, display its name, specify whether the status window displays when the Sync ActiveX runs, and execute the Sync ActiveX.

To specify the Sync data file You can add a command button that when clicked displays a Select Sync Data File dialog box in which a user can navigate to and select the Sync data file that the ActiveX will run. The script for the command button's Clicked event is:

```
string docname, named
integer value
value = GetFileOpenName("Select Sync Data File", &
+ docname, named, "SYC", &
+ "SYC Files (*.SYC),*.SYC," &
+ "All Files (*.*),*.*")
IF value = 1 THEN Ole_1.object.SyncFileName = docname
```

To specify whether the status window displays You can add a checkbox to the window that lets a user specify that a status window displays when the Sync ActiveX runs. The script for the checkbox's Clicked event is:

```
IF
This.Checked THEN
Ole_1.object.ShowStatus=TRUE
ELSE
```

```
Ole_1.object.ShowStatus=FALSE  
END IF
```

To display the current Sync data file You can add a command button to the window that when clicked displays the current Sync data file in static text. The script for the button's Clicked event is:

```
st_1.Text = Ole_1.Object.SyncFileName
```

To execute the Sync ActiveX and display a value You can add a command button that when clicked executes the Sync ActiveX and displays a message in static text. The script for the button's Clicked event is:

```
boolean rc  
rc = Ole_1.Object.Execute()  
IF rc  
THEN  
    st_2.text = "Execute Returned True"  
ELSE  
    st_2.text = "Execute Returned False"  
END IF
```

What's next

For information about how to deploy an HTML page containing the Sync ActiveX and an application with the Sync ActiveX in a window, see Chapter 5, "Deploying Synchronizing Applications and HTML Pages".

Deploying Synchronizing Applications and HTML Pages

About this chapter

This chapter is about deploying applications that synchronize.

Contents

| Topic | Page |
|---|-------------|
| About deploying applications and Web pages that synchronize | 44 |
| Deploying an application that synchronizes its files | 45 |
| Deploying an HTML page containing the Sync ActiveX | 47 |
| Deploying an application with the Sync ActiveX in a window | 48 |
| Additional files that may be needed on users' systems | 49 |

About deploying applications and Web pages that synchronize

In addition to the deployment issues you normally consider after you create a PowerBuilder Windows application, you need to consider new deployment issues if you want:

- ◆ Deployed applications to synchronize before they run
- ◆ Web pages containing the Sync ActiveX to synchronize
- ◆ Windows containing the Sync ActiveX to synchronize

Deploying the Sync runtime executable

When you set up your deployed applications to synchronize, you will be deploying the Sync runtime executable.

No restrictions or fees

There are no restrictions or fees when you distribute the Sync runtime executable.

Deploying on both 16- and 32-bit systems

Sync data files that the 16-bit Sync runtime executable runs must reference directories that have short filenames. So you may want to set up directories with short filenames on your network and always reference these directories when you create a Sync data file. Then the Sync data file can be used when you deploy both 16- and 32-bit applications.

Deploying an application that synchronizes its files

Standard installation directories

In most large corporations, applications are installed in standard directories on the corporate network or users' systems. It's easiest to synchronize application files in an environment that has standard installation directories.

What you deploy

When you deploy an application that synchronizes its files, you deploy the following files:

| Files to deploy | Where to deploy the files |
|---|------------------------------|
| Sync runtime executable 32-bit: SYNCRTE.EXE 16-bit: SYNCRTE16.EXE | Network server |
| Windows 95 and NT 4.0: Shortcut file (LNK file) Windows 3.1 or NT 3.51: Group file that contains a program item (GRP file) | Network server |
| Sync data file that has file copying instructions and an instruction to start the application | Network server |
| Application source files that Sync will copy to the destinations | Network server or FTP server |

After you deploy the required files, you may need to setup additional files on users' systems.

FOR INFO For information, see "Additional files that may be needed on users' systems" on page 49.

How you deploy

Use the following steps to deploy an application that synchronizes its files.

❖ **To deploy an application that synchronizes its files:**

- 1 Deploy the Sync runtime executable, the Sync data file, and the source files that Sync will synchronize on the network.
- 2 Depending on your users' operating systems, create a shortcut or a new group containing a program item that points to the Sync runtime executable on the network and executes instructions in the Sync data file on the network.

For example, in the Shortcut tab of the shortcut's properties, the following Sync command points to the Sync runtime executable and a Sync data file in a Sync directory on the network:

```
\\serv\c\sync\syncrt.exe \\serv\c\sync\syncpb.syc
```

Using the path in the command rather than a mapped drive is better. If you use a mapped drive (for example I:), all users must use the same letter for the mapped drive or the shortcut won't work.

- 3 (Optional) Change the shortcut or program item icon from the Sync icon to your PowerBuilder application's icon.
- 4 Put the shortcut or group file containing the program item on the network in a place accessible to all users.
- 5 Tell users how to use this shortcut or program item to run the application. How and where they use the shortcut or program item depends on your implementation:

- ◆ Users can use the shortcut or program item on the network.
- ◆ Users can copy the shortcut file or group file containing the program item to their systems and use it there.

For example, a user can copy the shortcut to the Windows Start Menu directory and/or the Windows Desktop directory.

- ◆ You can include an instruction in the Sync data file to copy the shortcut or group file containing the program item to users' systems. Then the first time users run the Sync runtime executable, they run it on the network and the shortcut or program item is copied. Then they can use the shortcut or program item on their systems.

For example, an instruction in the Sync data file could copy the shortcut to the Windows Start Menu directory and/or the Windows Desktop directory.

What users do

Users start the PowerBuilder application using the shortcut or program item on the network or on their system. First Sync executes to synchronize application files, and then Sync starts the application.

Deploying an HTML page containing the Sync ActiveX

What you deploy

After you create an HTML page and insert the Sync ActiveX in the HTML page, you deploy the following files:

| Files | Where to deploy the files |
|--|---|
| HTML page containing the Sync ActiveX | Web server directory that contains HTML pages |
| Sync data file | Network server or HTTP server |
| Source files that Sync will copy to the destinations | Network server or FTP server |

What you do

After you deploy the required files, you may need to set up additional files on users' systems.

FOR INFO For information, see "Additional files that may be needed on users' systems" on page 49.

What a user does

In a browser that supports ActiveX, a user accesses the Web page with the Sync ActiveX inserted.

When you created the HTML page, if you included the CodeBase property in the Object element, the Sync ActiveX is downloaded to the user's system and automatically registered and then the specified synchronization occurs. The Sync ActiveX is downloaded to the Windows Occache directory by default.

If you did not include the CodeBase property in the Object element, deploy SYNC.OCX and register it on each user's system using your site-specific method.

Deploying an application with the Sync ActiveX in a window

What you deploy

When you deploy an application that has a window in which the Sync ActiveX is inserted, you deploy the following files:

| Files | Where to deploy the files |
|--|--|
| Sync ActiveX | Include SYNC.OCX with the application files that you deploy to users when you build an installation for your application |
| Sync data file | Network server or HTTP server |
| Source files that Sync will copy to the destinations | Network server or FTP server |

What you do

After you deploy the required files, you may need to set up additional files on users' systems.

FOR INFO For information, see "Additional files that may be needed on users' systems" on page 49.

Installing and registering the Sync ActiveX

Normally you deploy the Sync Active and register it as part of your normal application deployment process. If you use an installation building tool such as InstallShield, you can specify the installation of the Sync ActiveX. A free InstallShield Version 5 is included with PowerBuilder. The Sync ActiveX is self registering and should be installed in the Windows Occache directory.

Additional files that may be needed on users' systems

When you deploy the Sync runtime executable (16- or 32-bit) and the Sync ActiveX, you may also need to set up additional files on users' systems.

For the 16-bit Sync runtime executable

The 16-bit Sync runtime executable requires Winsock files for TCP/IP on the user's system. Winsock files can't be freely distributed; you must purchase them.

For the 32-bit Sync runtime executable

The 32-bit Sync runtime executable requires Winsock files and Wininet files on the user's system. Windows 95 and NT has an installation option for installing Winsock files. Wininet files are installed when you install PowerBuilder, and they are also in the deployment kit.

The Wininet files are:

HLINK.DLL
HLINKPRX.DLL
INLOADER.DLL
OLEAUT32.DLL
STDOLE2.TLB
URLMON.DLL
WININET.DLL

For the Sync ActiveX

The Sync ActiveX requires Wininet files and Microsoft Foundation Class (MFC) Runtime files on the user's system.

The Wininet files are:

HLINK.DLL
HLINKPRX.DLL
INLOADER.DLL
OLEAUT32.DLL
STDOLE2.TLB
URLMON.DLL
WININET.DLL

The MFC Runtime files are:

MFC42.DLL
MFVCIRT.DLL
MSVRT.DLL
OLEPRO32.DLL

The MFC Runtime files are usually in the Windows 95 or Windows NT system directory. If users run Microsoft Internet Explorer, they already have the MFC Runtime files.

Sync Data Files

About this chapter

This chapter describes the structure of Sync data files.

Contents

| Topic | Page |
|--------------------------|-------------|
| About Sync data files | 52 |
| Sync data file structure | 53 |
| [Sync] section | 54 |
| [Log] section | 56 |
| [Start] section | 57 |
| [Variables] section | 58 |
| [End] section | 58 |

About Sync data files

Sync gets its instructions from a Sync data file. Sync data files can have any extension; the default extension is SYC.

Text file

A Sync data file is an ordered text file that looks like an INI file. A Sync data file contains sections and keywords that specify two basic types of instruction:

- ◆ Copy a source file to a target destination to synchronize source and target files and provide an address for another Sync data file for instructions
- ◆ (Optional) Execute an application after the synchronization process completes so application files can be updated before the application starts

Other items

Other optional sections and keywords provide additional synchronization features such as logging and displaying a status window.

Example

Here's a sample Sync data file:

```
[LOG]
file=D:\Panther\Sync32\results.txt
replace=0
verbose=3
[SYNC]
syncop0=/src D:\Panther\Sync32\*.syc /dest
c:\syncdest\sync drivefile\ /d /s /v
syncop1=/src \\houdevel\pbdev2\examples\*.bmp /dest
c:\syncdest\sync driveunc\ /d
syncop2=/src ftp:\\ftp.powersoft.com\pub\info\*.txt
/dest c:\syncdest\ftpdest\ /d
[VARIABLES]
logfile=D:\Panther\Sync32\results.txt
[END]
exec=C:\WINDOWS\notepad.exe
parm=%logfile%
[START]
show=1
cancel=0
```

Sync data file structure

A Sync data file is a structured file that has one required section and four optional sections, which can be in any order:

| Section | Purpose of instructions in the section |
|----------------------|---|
| [Sync] (required) | Compare master files with source files and copy master files from the source to the target |
| [Log] | Record the operations that take place during the synchronization process with the detail you specify |
| [Start] | Display a status window (with or without a Cancel button) so the user has visual feedback that synchronization is happening |
| [Variables] | Specify runtime variable names and values for sources, destinations, log files, applications, or command-line parameters |
| [End] | Run an application after the synchronization process completes |

[Sync] section

Purpose When Sync executes instructions in the [Sync] section, Sync compares source files with destination files and copies files from the source to the destination. The [Sync] section can also contain a reference to another Sync data file.

Keywords The [Sync] section has keywords that provide the synchronization instructions to Sync that copy source files to destinations:

| Keyword | Description |
|----------------|------------------------------------|
| Syncop0 | First synchronization instruction |
| Syncop1 | Second synchronization instruction |
| SyncopN | N+1st synchronization instruction |

SyncopX keywords must be ordered from 0 to N.

File comparison methods Sync can compare source and destination files in three ways to determine whether to copy source files to the destinations:

| Comparison method | Parameter used for this method |
|---|---------------------------------------|
| DateTime stamps | /D |
| Sizes | /S |
| Version information embedded in EXEs and DLLs | /V |

For FTP sources

Version checking works for local or network drives, but not for FTP. DateTime checking is against the date only.

For example:

```
[Sync]
;Copy all txt files in rfc directory on
;Microsoft.com using check for dates
Syncop0=
/SRC ftp:\\ftp.microsoft.com\developr\rfc\*.txt
/DEST c:\syncdest\ /D
;copy all text files in info directory on
;powersoft.com using check for size
syncop1=
/SRC ftp:\\ftp.powersoft.com\pub\info\*.txt
```

```
/DEST c:\syncdest\ftpdest\ /S
```

Nesting a Sync data file

A Sync data file can have another Sync data file nested in it. To nest a Sync data file, you provide a reference to the location of the nested Sync data file in the [Sync] section.

For example, the last instruction in the following [Sync] section provides the location of a nested Sync data file:

```
[Sync]
;Copy all txt files in rfc directory on
;Microsoft.com using check for dates
Syncop0=
/SRC ftp:\\ftp.microsoft.com\developr\rfc\*.txt
/dest c:\syncdest\ /d
;nested sync
Syncop1=I:\SyncFiles\dptsync.sync
```

[Log] section

Purpose When Sync executes instructions in the [Log] section, Sync records the operations that take place during the synchronization process in a log file with the detail you specify.

Keywords The [Log] section has keywords that make the logging of the execution of Sync data file instructions occur. Each time that logging occurs, the log file can be replaced or new logging statements can be appended to the existing file. Sync will create a specified log file if it does not exist.

The [Log] section can have these keywords:

| Keyword | Description |
|----------------|---|
| File | Location and name of a log file If there is no existing log file, the specified log file will be created |
| Replace | Replace or append to an existing log file: 0 Append (default) 1 Replace |
| Verbose | Detail level of the log file: 1 Errors only (default) 2 Normal 3 Verbose |

For example:

```
[Log]
File = c:\windows\desktop\mysync.log
Replace = 1
Verbose = 2
```

The logging instructions will display in the status window if the [Start] section has an instruction to display a status window.

[Start] section

Purpose When Sync executes instructions in the [Start] section, Sync displays a status window (with or without a Cancel button) so the user has visual feedback that synchronization is happening.

Keywords The [Start] section can have two keywords:

| Keyword | Description |
|----------------|---|
| Show | Display a status window: |
| | 0 Don't display |
| | 1 Display (default) |
| Cancel | Include a Cancel button on the status window: |
| | 0 Don't include (default) |
| | 1 Include |

For example:

```
[start]
show = 1
cancel = 0
```

[Variables] section

Purpose When Sync executes instructions in the [Variables] section, Sync uses the values of the runtime variables you define as sources, destinations, log files, applications, or command-line parameters.

Keywords The [Variables] section can have any number of keywords that are runtime variables. You define the variable names and their values in this section.

For example:

```
[Variables]
Variable1 = Value1
Variable2 = Value2
```

Defining a runtime variable

A runtime variable can be used to specify a:

- ◆ Source
- ◆ Destination
- ◆ Log file
- ◆ Application that runs after synchronization completes
- ◆ Command-line parameter associated with the application

For example, you may want to define a runtime variable with the name `INSTALLDIR` that refers to a very long directory name you are using for a destination. For example, you might define a runtime variable for `C:\APPLICATIONS\PROGRAMS\AD\POWERBUILDER\`.

The runtime variable `InstallDir` is defined by the following lines in the `SYC` file:

```
[Variables]
InstallDir = C:\applications\programs\ad\PowerBuilder\
```


[End] section

Purpose When Sync executes instructions in the [End] section, Sync runs an application. This instruction is always executed after the synchronization process completes.

Keywords The [End] section has keywords that cause an application to start after all other synchronization instructions have executed. You can pass command-line parameters for the application you are running.

The [End] section can have two keywords:

| Keyword | Description |
|----------------|--|
| Exec | Application to run after sync instructions are completed |
| Parm | One or more command-line parameters for the application you are running. Normally you'll have no command-line parameters, but there may be cases where you want to use them for a particular program |

For example, the following instructions open Microsoft Word, and then Word opens an RTF file on the network and a local DOC file:

```
[end]
Exec = c:\msoffice\winword\winword.exe
Parm = i:\dept\file.rtf c:\local\file.doc
```

[End] section

Index

A

- AboutBox() method of Sync ActiveX 31
- ActiveX Control Pad 32
- ActiveX-friendly editors 32
- adding an instruction
 - to copy files 13
 - to define a runtime variable 17
 - to display a status window 19
 - to log synchronization 17
 - to nest a Sync data file 15
 - to run an application 16
- allowing ActiveXs to write to hard disk 37
- applications that synchronize
 - additional files users may need 49
 - deploying 45
- applications with Sync ActiveX in a window
 - creating 38
 - deploying 48

B

- BODY tag in HTML 33, 35
- browser security settings 37

C

- canceling synchronization 20
- changing security settings in browser 37
- code examples for the Sync ActiveX in a window 39
- coding HTML 32
- command format for the Sync runtime executable 25, 26
- Command Properties dialog box 13, 15
- comparison methods for copying files 14
- components of Sync 2
- copying an instruction 12
- copying files 14

- creating a Sync data file 11
- cutting instructions 12

D

- date and time checking of files 14
- defining runtime variables 18
- deleting
 - instructions 12
 - runtime variables 19
- deploying
 - applications that synchronize 44
 - applications with Sync ActiveX in a window 48
 - Sync runtime executable 44
 - Web pages that synchronize 47
- destinations
 - for Sync to copy files to 9
 - ways to specify 14
- detail level for logging 17
- displaying a status window 19
- distributing the Sync runtime executable 44

E

- editing
 - runtime variables 19
 - synchronization instructions 12
- editors, ActiveX-friendly 32
- enabling ActiveXs in a Web browser 37
- End section in Sync data file 59
- Execute() method of Sync ActiveX 31
- executing the Sync ActiveX in HTML 35

F

- file comparison methods 14
- file version checking 14

files for Sync to copy, specifying 13
files needed for Sync 5
FTP servers
 as sources 8
 nested Sync data files 15
FTP sources, file comparison methods 14

H

HTML
 coding 32
 deploying pages that synchronize 47

I

inserting the Sync ActiveX
 in a Web page 33
 in an application window 38
installation directories 45
installing Sync 5
InstallShield 48

J

JavaScript OnLoad event handler 35

L

local drives as sources and destinations 8
location
 for nested Sync data files 15
 for Sync data file for Sync ActiveX 30
log file, level of detail in 17
Log section in Sync data file 56
logging the synchronization process 17

M

mapped network drives 25, 26
methods of the Sync ActiveX 31
Microsoft ActiveX Control Pad 32

Microsoft Foundation Class runtime modules
 installation 5
 setup on users' systems 49
Microsoft Front Page 32

N

nesting Sync data files 15
network drives as sources and destinations 8

O

OCX properties in PowerBuilder 38
OnLoad event handler in HTML 35
opening a Sync data file 11
optional parameters 25
Options dialog box
 Log File tab 17
 Runtime Variables tab 18
 Start and End Options tab 16

P

passing a reference to a Sync data file 25
pasting an instruction in Sync Builder workspace 12
path of network drive 25
PowerBuilder window
 code examples for the Sync ActiveX 39
 inserting the Sync ActiveX 38
properties of the Sync ActiveX
 about 31
 specifying in HTML 34
purpose of Sync 2

R

registering the Sync ActiveX 48
required files 5
running
 application after synchronization 16
 Sync runtime executable 25, 26, 27

- runtime variables
 - in copy instructions 13
 - in nested Sync data files 15
 - system-defined 17
 - user-defined 17

- S**
- sample Sync data file 52
- security settings in browser 37
- short filenames 8
- showing a status window 19
- ShowStatus property of Sync ActiveX 31
- size checking of files 14
- sources
 - for Sync to copy files 8
 - ways to specify 14
- specifying runtime variables 18
- standardization for installation directories 45
- Start section in Sync data file 57
- starting Sync Builder 10
- status window, displaying
 - for Sync Active X in a Web page 35
 - for Sync Active X in a window 38
- steps
 - for creating a Sync data file 11
 - for using Sync 4
- structure of Sync data files 53
- Sync
 - about 2
 - components 2
 - installing 5
 - purpose of 2
 - required files 5
 - sources and destinations 8
 - starting Sync Builder 10
 - steps for using 4
 - what it does 2
- Sync ActiveX 30
 - changing security settings in browser 37
 - inserting in a Web page 33
 - installation 5
 - location of Sync data file 30
 - properties and methods 31
 - registration of 48
 - specifying execution of 35
 - specifying properties 34
 - testing a Web page containing it 36
 - using in a Web page 32
 - using in a window 38
- Sync Builder
 - adding an instruction to copy files 13
 - cancelling the synchronization process 20
 - copying an instruction in workspace 12
 - creating a Sync data file 11
 - cutting an instruction in workspace 12
 - defining runtime variables 17
 - deleting an instruction in workspace 12
 - deleting runtime variables 19
 - displaying a status window 19
 - editing an instruction in workspace 12
 - editing runtime variables 19
 - logging synchronization 17
 - nesting a Sync data file 15
 - opening a Sync data file 11
 - pasting an instruction in workspace 12
 - running an application 16
 - sources and destinations 8
 - starting 10
 - testing Sync data files 21
 - using short filenames 8
- Sync data files
 - about 52
 - adding an instruction to copy files 13
 - copying an instruction 12
 - creating 11
 - cutting an instruction 12
 - defining runtime variables 17
 - deleting an instruction 12
 - deleting runtime variables 19
 - displaying a status window 19
 - editing an instruction 12
 - editing runtime variables 19
 - End section 59
 - Log section 56
 - logging synchronization 17
 - nesting another Sync data file 15
 - opening 11
 - passing a reference to 25
 - pasting an instruction 12
 - running 21

- running an application 16
- short filenames 8
- Start section 57
- structure 53
- Sync section 54
- testing 21
- Variables section 58

Sync runtime executable

- about 24
- passing reference to Sync data file 25
- required files 49
- requirements for distributing 44
- run by associating SYC files with 26
- run by passing a command-line argument 27
- run by passing a reference to Sync data file 26
- sources and destinations 8

Sync section in Sync data file 54

SyncFileName property of Sync ActiveX 31

Synchronizer *see* Sync

system-defined runtime variables 18

T

testing

- Sync data files 21
- Web pages that have the Sync ActiveX 36

time checking of files 14

U

user-defined runtime variables 17

using runtime variables 19

using the Sync ActiveX

- in a Web page 32
- in an application window 38

V

valid ways to specify copying files 14

variables *see* runtime variables

Variables section in Sync data file 58

W

Web browser security settings 37

Web pages that synchronize

- creating 32
- deploying 47

Wininet files

- installation 5
- setting up on users' systems 49

Winsock files 49