

New feature in PB 11.1

- [Vista support](#)
- [Incremental rebuild for .NET application targets](#)
- [RadControl preview for Web Forms applications](#)
- [OrcaScript enhancements](#)
- [Pipeline support in Windows Forms](#)
- [Web service client support in Web Forms](#)
- [RightToLeft support for DatePicker and MonthCalendar](#)
- [Page display properties for RichTextEdit control](#)
- [Specifying null values as retrieval arguments](#)
- [Informix 10 support](#)
- [Statement caching for Oracle 10g](#)
- [Identity columns in Adaptive Server 15](#)

[Vista support](#)

The PowerBuilder 11.1 development environment has been tested on the Microsoft Vista operating system using both the 32-bit and 64-bit versions of the Vista Business Edition. Application deployment has been tested using the same Vista editions. The PowerBuilder development environment and applications created using PowerBuilder run as 32-bit applications on both 32-bit and 64-bit platforms.

Microsoft has established two separate levels of its Windows Vista Logo Program for Software: "Certified for Windows Vista" and "Works with Windows Vista". PowerBuilder 11.1 satisfies the Works with Windows Vista requirements. Applications built with PowerBuilder can be designed to meet the Certified for Windows Vista requirements. For more information about these requirements, see the Microsoft Web site.

- [User accounts and administrative privileges](#)
- [Installing PowerBuilder](#)
- [Running PowerBuilder](#)
- [Manifest files](#)
- [Signed DLLs](#)
- [Setting up an application pool for Web Forms](#)
- [IIS 6 compatibility component required](#)
- [Setting up permissions for Web Forms applications](#)
- [Windows Help files](#)
- [EAServer support](#)

- [Limitations on Windows Vista](#)

[User accounts and administrative privileges](#)

One of the most significant changes in Windows Vista is the introduction of User Account Control (UAC). UAC is the mechanism Vista uses to limit the default privileges users run with on Vista. Limiting these privileges is intended to make it less likely that a user's actions affect the configuration of the system or other users' state and settings.

Standard users (members of the Users group) have restricted privileges. They cannot install most applications, change system settings, or perform most other administrative tasks.

Administrators have elevated privileges. When a user who is a member of the Administrators group (or another group with elevated privileges) logs on to a computer running Vista, the interactive login process creates a restricted token in addition to the token that describes the administrator's elevated privileges. The restricted token has the privileges of a standard user and is used to create the interactive desktop and Windows Explorer session.

The elevated privileges for the user are still available when needed. If the user attempts to perform an operation that requires administrative privileges, the user is prompted to authorize the operation.

When the user double-clicks an icon or selects it from the Start menu, the process inherits the restricted token. If the application is marked as requiring elevated privileges in a manifest file, UAC prompts the user for authorization to run with those privileges. For more about manifest files, see ["Manifest files"](#).

You can start an application with administrative privileges by right-clicking it in the Start menu or Explorer and selecting Run As Administrator from its pop-up menu.

[Installing PowerBuilder](#)

You must have administrative privileges to install PowerBuilder on Windows Vista. To ensure that the installation completes successfully, right-click the setup.exe or autorun.exe file and select Run as administrator.

Running PowerBuilder

PowerBuilder typically writes to multiple files in the *Program Files* directory, including *pb.ini*, *PB110_sysam.properties*, and code examples, tutorial, and demo database files. If an application is run without administrative privileges, Vista restricts it from writing to the *HKEY_LOCAL_MACHINE* subtree in the registry, the *Program Files* directory, and the Windows directory. Instead, changes are written to a writable area in the registry and to the user's local directory. This is referred to as virtualization.

For example, if you start PowerBuilder without administrative privileges, the *pb.ini* file is written to *Program Files\Sybase\PowerBuilder 11.0\pb.ini* in the *C:\Users\\AppData\VirtualStore* directory.

To avoid the issues that would result from this behavior, on Vista, you must start PowerBuilder 11.1 with administrative privileges by right-clicking *pb110.exe* in the Start menu or Explorer and selecting Run as administrator from its pop-up menu.

Manifest files

Manifest files

If you want to deploy an application to the Windows Vista operating system that meets the certification requirements of the Windows Vista Logo program, you must follow User Account Control (UAC) guidelines. The executable file must have an embedded manifest that defines the execution level and specifies whether access to the user interface of another window is required. The Vista Application Information Service (AIS) checks the manifest file to determine the privileges with which to launch the process. Use the Security tab page in the Application and .NET Windows Forms Project painters to specify those properties.

Generate options

Select Embedded manifest if your application needs to be certified for Vista. A manifest file with the execution level you select is embedded in the application's executable file.

You can also select External manifest to generate a standalone manifest file in XML format that you ship with your application's

executable file, or No manifest if you do not need to distribute a manifest file.

.NET 2.0 SDK required for embedded manifest If you select Embedded manifest for a Windows Forms target, you must have the .NET 2.0 SDK installed on your system, because the process that embeds the manifest in the executable file uses the **mt.exe** tool that is distributed with the SDK.

Execution level

Select As Invoker if the application does not need elevated or administrative privileges. Selecting a different execution level will probably require that you modify your application to isolate administrative features in a separate process to receive Vista certification.

Select Require Administrator if the application process must be created by a member of the Administrators group. If the user did not start the process as an administrator, a message box displays so that the user can enter the appropriate credentials.

Select Highest Available to have the AIS retrieve the highest available access privileges for the user who started the process.

UI access

If the application needs to drive input to higher privilege windows on the desktop, such as an on-screen keyboard, select the Allow access to protected system UI check box. For most applications you should not select this check box. Microsoft provides this setting for user interface Assistive Technology (Section 508) applications.

Authenticode signing required If you check the Allow access to protected system UI box, the application must be Authenticode signed and must reside in a protected location, such as Program Files or Windows\system32.

Warning - Vista manifest files may crash XP If you add a manifest file that contains Vista extensions to an executable or DLL file on Windows XP with Service Pack 2, memory may be corrupted, causing the operating system to crash or restart. This is a known Microsoft

issue described on the Microsoft support site at <http://support.microsoft.com/Default.aspx?kbid=921337> .

To resolve this issue on Windows XP, Microsoft has provided a patch that can be downloaded from the Microsoft Web site at <http://www.microsoft.com/technet/security/Bulletin/MS06-075.msp> [x](#) . You should install this patch before running PowerBuilder on XP.

Signed DLLs

When you run an application on Vista, Vista warns you if the application uses any DLLs that have not been signed with an Authenticode certificate. If you want to deploy your application with Windows Vista Logo certification, all the DLLs that you distribute with it must be signed. PowerBuilder runtime DLLs, and all other PowerBuilder DLLs, are signed.

Setting up an application pool for Web Forms

Virtual directories in IIS are hosted in an application pool. An application pool is the host process for one or more Web applications. When you deploy a PowerBuilder Web Forms application to IIS 7, the application is deployed to the DefaultAppPool. By default, this application pool uses the Integrated managed pipeline mode, where managed code is integrated into the request processing pipeline. To avoid compatibility issues, Web Forms applications deployed from PowerBuilder must run in an application pool that uses the Classic managed pipeline mode, where ASP.NET runs as an ISAPI extension.

Although you can configure the DefaultAppPool to use the Classic mode, changing the setting will affect all applications running in that application pool. A better option is to create a new application pool for PowerBuilder Web Forms applications.

Creating a PowerBuilder Web Forms application pool

If you choose not to create a new application pool, you must complete step 4 in the following procedure for the application pool that hosts your application.

To create and configure an application pool for PowerBuilder Web Forms applications:

1. In IIS Manager, select Application Pools.
2. In the Actions pane, select Add Application Pool.
3. Provide a name, such as PBWebForms, for the application pool.
4. Set Managed Pipeline Mode to Classic and click OK.

The new application pool displays in the list.

Enabling 32-bit applications on 64-bit Vista

On 64-bit Vista, you must enable the application pool to run 32-bit applications.

To enable 32-bit applications on 64-bit Vista:

1. In IIS Manager, select Application Pools.
2. In the list of Application Pools, select the application pool you have configured for use with PowerBuilder Web Forms.
3. In the Actions pane, select Advanced Settings under Edit Application Pool.
4. Expand the General settings, set Enable 32-bit Applications to True, and click OK.

Moving an application into a different application pool

If you have created and configured a new application pool for PowerBuilder, you need to move your PowerBuilder Web Forms applications into the pool.

To move an application into a different application pool:

1. In IIS Manager, expand Web Sites and Default Web Site.
2. Right-click the virtual directory for your application and click Advanced Settings.

3. Select the drop-down list next to the Application Pool property, select the application pool you created, and click OK
4. Reload the application.

IIS 6 compatibility component required

If you deploy a Web Forms application to a remote server running IIS 7, or if you publish a smart client application to a local or remote server running IIS 7, the Vista Metabase Compatibility component of IIS 7 must be installed on the server. This component is not installed by default. It is not required to deploy a Web Forms application to a local server.

You can install it from the Programs and Features page in the Windows control panel. Select Turn Windows features on or off, then select Internet Information Services>Web Management Tools>IIS 6 Management Compatibility>IIS Metabase and IIS 6 configuration compatibility.

When you deploy a new Web Forms target, a temp directory is created in the *Inetpub\wwwroot\application_name* directory, where *application_name* is the name of your application, and several subdirectories are created in the *Inetpub\wwwroot\application_name_root* directory. Files are written to and deleted from these directories, therefore the IIS_IUSRS group must have full permissions on *temp* and *application_name_root*.

Before a PowerBuilder .NET Web Forms application connects to a SQL Anywhere® database, you must either start the database manually or grant the IIS_IUSRS group on IIS 7 default permissions for the *Sybase\Shared* and *Sybase\SQL Anywhere* directories, making sure to replace permissions of all child objects in those directories. Full permissions are required for the database and database log files used by your application.

If you do not grant the appropriate user permissions for Sybase directories and your database configuration is set to start the database automatically, your application will fail to connect to the database. SQL Anywhere cannot access files unless the IIS_IUSRS user group has the right to access them.

For more information, see the section on configuring ASP.NET in Deploying Applications and Components to .NET.

Setting up permissions for Web Forms applications

Windows Vista does not distribute the WinHlp32.exe file required to open Windows Help files such as the pbhlp110.hlp file used in PowerBuilder. To use .hlp files, you need to download a special Vista version of WinHlp32.exe from the Microsoft Web site at <http://go.microsoft.com/fwlink/?LinkID=82148> .

Compiled HTML Help (.chm) files are supported, but you need to edit the Windows registry to enable a Help macro that supports links from the pbhlp110.hlp file to the pbman110.chm file. If you do not edit the registry, the "For more information" links at the bottom of many topics in the Windows Help display an error.

You also need to edit the registry if you need to run Windows Help files at a remote location on an intranet.

Registry reflection on 64-bit Windows 64-bit versions of Windows use registry reflection to maintain a 32-bit registry view and a 64-bit registry view. On 64-bit Windows, configuration information related to 32-bit applications is stored in the HKEY_LOCAL_MACHINE\Software\WOW6432node registry hive.

To enable Windows Help macros and remote access on Vista:

1. Create the following registry key.

On 32-bit Windows:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WinHelp
```

On 64-bit Windows:

```
HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432node\Microsoft\Win  
Help
```

2. Add a new DWORD value with the name AllowProgrammaticMacros and the value 1.
3. Add a new DWORD value with the name AllowIntranetAccess and the value 1.

You can also add this support by saving the following lines in Notepad to a file with the extension .reg and importing it into the registry.

On 32-bit Windows:

Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WinHelp]

"AllowProgrammaticMacros"=dword:00000001

"AllowIntranetAccess"=dword:00000001

On 64-bit Windows:

Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432node\Microsoft\WinHelp]

"AllowProgrammaticMacros"=dword:00000001

"AllowIntranetAccess"=dword:00000001

Microsoft prohibits the distribution of *WinHlp32.exe* with deployed applications. If your application uses *.hlp* files, you should provide your users with instructions on how to download *WinHlp32.exe*. For more information, see the Microsoft support site at <http://support.microsoft.com/kb/917607> .

Windows Vista does not distribute the *WinHlp32.exe* file required to open Windows Help files such as the *pbhlp110.hlp* file used in PowerBuilder. To use *.hlp* files, you need to download a special Vista version of *WinHlp32.exe* from the Microsoft Web site at <http://go.microsoft.com/fwlink/?LinkID=82148> .

Compiled HTML Help (*.chm*) files are supported, but you need to edit the Windows registry to enable a Help macro that supports links from the *pbhlp110.hlp* file to the *pbman110.chm* file. If you do not edit the registry, the "For more information" links at the bottom of many topics in the Windows Help display an error.

You also need to edit the registry if you need to run Windows Help files at a remote location on an intranet.

Registry reflection on 64-bit Windows 64-bit versions of Windows use registry reflection to maintain a 32-bit registry view and a 64-bit registry view. On 64-bit Windows, configuration information related to 32-bit applications is stored in the `HKEY_LOCAL_MACHINE\Software\WOW6432node` registry hive.

To enable Windows Help macros and remote access on Vista:

1. Create the following registry key.

On 32-bit Windows:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WinHelp
```

On 64-bit Windows:

```
HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432node\Microsoft\Win  
Help
```

2. Add a new DWORD value with the name AllowProgrammaticMacros and the value 1.
3. Add a new DWORD value with the name AllowIntranetAccess and the value 1.

You can also add this support by saving the following lines in Notepad to a file with the extension *.reg* and importing it into the registry.

On 32-bit Windows:

```
Windows Registry Editor Version 5.00
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WinHelp]
```

```
"AllowProgrammaticMacros"=dword:00000001
```

```
"AllowIntranetAccess"=dword:00000001
```

On 64-bit Windows:

```
Windows Registry Editor Version 5.00
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432node\Microsoft\WinHelp]
```

```
"AllowProgrammaticMacros"=dword:00000001
```

```
"AllowIntranetAccess"=dword:00000001
```

Microsoft prohibits the distribution of *WinHlp32.exe* with deployed applications. If your application uses *.hlp* files, you should provide your users with instructions on how to download *WinHlp32.exe*. For more information, see the Microsoft support site at <http://support.microsoft.com/kb/917607> .

[EAServer support](#)

The Vista operating system is supported in EAServer 6.0.2 and later.

Limitations on Windows Vista

This section lists some limitations of the Vista operating system that affect PowerBuilder:

- [JSP targets](#)
- [Controls that use calendars](#)
- [Saving as PDF](#)

For additional issues, see the "Known Issues" section in the PowerBuilder Release Bulletin.

JSP targets

On the Vista operating system, you can create a JSP target and a JSP page, but the component used to implement the HTML Editor's Page view and its built-in Script editor is not supported on the Vista operating system, therefore JSP targets are not supported on Vista.

Controls that use calendars

The Vista operating system does not support several properties for the DatePicker, EditMask, and MonthCalendar controls and the drop-down calendar in a DataWindow column. The following properties are not supported on Vista:

- DatePicker: CalendarBackColor, CalendarFontName, CalendarFontWeight, CalendarItalic, CalendarTextColor, CalendarTextSize, CalendarTitleBackColor, CalendarTitleTextColor, CalendarTrailingTextColor, CalendarUnderLine
- EditMask: CalendarBackColor, CalendarTextColor, CalendarTitleBackColor, CalendarTitleTextColor, CalendarTrailingTextColor
- MonthCalendar: FaceName, MonthBackColor, TextColor , TextSize, TitleBackColor, TitleTextColor, TrailingTextColor, Underline
- Column controls in DataWindow objects with a drop-down calendar EditMask style: DDCal_BackColor, DDCal_TextColor, DDCal_TitleBackColor, DDCal_TitleTextColor, DDCal_TrailingTextColor

In addition, the Vista operating system does not support the WeekNumbers property for the DatePicker control. When this property is true, the DatePicker control is not displayed correctly. The

same limitation applies to the MonthCalendar control when WeekNumbers is true and Autosize is false.

[Saving as PDF](#)

To save as PDF in the development environment, you must be running PowerBuilder as an administrator (see "[Running PowerBuilder](#)") and the correct versions of Microsoft PostScript printer driver files must be available in the drivers directory when a file is saved as PDF using the Distill method.

If you (and your users) have installed a PostScript printer, the PostScript driver files required to create PDF files are already installed. Typical locations are *C:\WINDOWS\system32\spool\drivers\w32x86* on Windows XP,

C:\Windows\System32\DriverStore\FileRepository\ntprint.inf_XXXXXxxx, where *XXXXXXXX* is a system-specific identifier, in the *1386* subdirectory on 32-bit Vista and the *Amd64* subdirectory on 64-bit Vista. You must use the version of these files that is appropriate to the system where the PDF file is created. Copy the files to the *Shared\PowerBuilder\drivers* directory.

If you have never installed a PostScript printer, you can use the Printers and Faxes option in the Windows control panel to install a generic PostScript printer. If the Microsoft *PSCRIPT5.DLL* has never been installed, you may be prompted to insert the Windows Vista install CD.

[Incremental rebuild for .NET application targets](#)

PowerBuilder 11.1 introduces incremental builds for .NET application targets, allowing you to save time while deploying applications for testing or production purposes. This feature means that only the classes that are affected by one or more changes are recompiled during the build process.

Target level

The incremental rebuild process for .NET targets is conducted as the first step of a project's deployment to a .NET platform. Although deployment remains at the project level, incremental rebuilds are done at the target level. This means multiple projects within a single target can benefit by sharing the same incremental build assemblies or .NET modules.

.NET Web Service and .NET Assembly targets Incremental builds are not available for .NET component targets. The PowerBuilder .NET compiler always does full rebuilds for these target types.

For detailed information on build directories and .NET modules created for incremental builds, see the chapter on "Compiling and Debugging" in the Deploying Applications and Components to .NET book, available in the compiled HTML Help.

Triggering build and deploy operations

PowerBuilder lets you trigger build and deploy operations when you run or debug a .NET Web Forms or Windows Forms project. By default, when you click the running man or debugging icon in the PowerBuilder toolbar, or select Run from a project menu or context menu for one of these target types, PowerBuilder determines if there is a corresponding build directory for the selected target. If there is, PowerBuilder checks whether the .NET modules in the build directory are consistent with the latest changes to each object in your current application.

If implementation or interface changes are detected or if the build directory does not exist for the current target, PowerBuilder displays a message box that tells you the project is out of date and that prompts you to redeploy the project. The message box has three buttons (Yes, No, and Cancel) and a check box that lets you prevent the display of the message box the next time you click or select run (or debug).

If you click Yes in the message box, PowerBuilder builds the project using an incremental or full rebuild—depending on the current rebuild scope—and then redeploys it, using the current project's deployment specifications. If you click No in the message box with the redeployment prompt, PowerBuilder attempts to run or debug the currently deployed target even though it is out of date. Clicking Cancel terminates the run or debug request.

If you select this "Do not ask me again" check box and then click the Yes or No button, PowerBuilder modifies a drop-down list selection on the General tab of the System Options dialog box.

System option

The "On click Run, if .NET application projects are out of date" drop-down list selection on the General tab of the System Options dialog box determines whether a message box displays when you run or debug a project and project objects have been modified. The following table shows the choices available in this drop-down list:

Selection	Effect when you click or select Run (or debug)
Ask me	(Default selection.) Causes a message box to display if the current project has been modified since the last time it was deployed or if it has never been deployed before.
Always redeploy	Always redeploys a project before running or debugging it. It first rebuilds the project using the rebuild scope set in the Project painter.
Never redeploy	Never redeploys a project before trying to run it, although it does deploy a project that has not been previously deployed, and then attempts to run or debug that project. (You should not use this option if you want to debug a project that you have previously deployed.)

The message box that prompts you to redeploy an out-of-date project can display only when the drop-down list selection is "Ask me." This selection changes automatically to "Always redeploy" if you click Yes in the message box when the "Do not ask me again" check box is selected. It changes to "Never redeploy" if you click No in the message box when the "Do not ask me again" check box is selected.

[RadControl preview for Web Forms applications](#)

The PowerBuilder 11.1 setup program installs Telerik RadControls for ASP.NET, and PowerBuilder automatically deploys these controls with your Web Forms applications. RadControls provide enhanced functionality for Web Forms toolbars and menus, and can be used in place of the IE Web Controls that were required for PowerBuilder 11.0. The PowerBuilder installed RadControl set also includes DatePicker and MonthCalendar controls.

Incomplete control set for RadControl preview The RadControl TreeView control is not included in the current PowerBuilder release, so if your Web Forms application uses TreeView controls, you must install IE Web Controls.

The PBWebControlSource global property lets you switch between "old style" controls and the installed RadControl control set. You can

set this property at design time on the Configuration tab of the Web Forms Project painter, or you can change its value after deployment in the Web.config file for your Web Forms application.

You can set one of the following values for the PBWebControlSource global property:

- **IE** This setting causes the Web Forms application to use IE Web Controls for application toolbars and menus, and "old style" controls for DatePicker and MonthCalendar controls. This is the default setting. You must install IE Web Controls for use with this setting.
- **RAD** This setting causes the Web Forms application to use RadControls for application toolbars and menus, and for DatePicker and MonthCalendar controls. If you use this setting with applications that include TreeView controls, you must still install IE Web Controls.

The ability to switch to IE Web Controls is maintained for backward compatibility only and should not be used for new development. Microsoft no longer provides support for these controls.

For information about installing IE Web Controls, see the first chapter in the Deploying Applications and Components to .NET book.

[OrcaScript enhancements](#)

Support for Unicode command files

In PowerBuilder 11.1, you can use either Unicode or ANSI command files. Previously, only ANSI command files were supported.

Building Windows Forms projects

PowerBuilder 11.1 includes a new ORCA library function for silently building and deploying a Windows Forms project according to the specifications contained in the project object.

For information on the PBORCA_DeployWinFormProject function, see the PDF file for the ORCA Guide that the PowerBuilder 11.1 setup program installs in the PowerBuilder *SDK\ORCA* directory.

You can call the Windows Forms deployment function with the following OrcaScript command:

```
Deploy WinForm Project "pbIName" "entryName" ["iconName"]
```

The arguments you supply for this command are the name of the library containing the project you want to deploy, the name of the project object, and optionally, the name of an application icon.

For more information about OrcaScript, see Appendix B in the PowerBuilder User's Guide.

[Pipeline support in Windows Forms](#)

PowerBuilder 11.1 supports the use of Pipeline objects in Windows Forms targets. This support is in a new DLL, *pbdp110.dll*, which must be deployed with both .NET Windows Forms and standard PowerBuilder applications. The DLL has been added to the Runtime Packager.

[Web service client support in Web Forms](#)

You can use the built-in Web services client extension (*pbwsclient110.pbx*) in both .NET Web Forms and Windows Forms targets in this release. You cannot use any other PBNI extensions in a .NET target.

[RightToLeft support for DatePicker and MonthCalendar](#)

PowerBuilder 11.1 supports the RightToLeft property in the DatePicker and MonthCalendar controls. The property is not supported in these controls in Web Forms targets.

[Page display properties for RichTextEdit control](#)

Page display properties for RichTextEdit controls allow you to manage the size and orientation of documents inside the controls. These display properties are separate from the page properties that can be set at runtime for printing the contents of RichTextEdit controls. They are also separate from the height and width properties of the controls themselves.

The following table lists and describes the new display properties. These properties are implemented for standard PowerBuilder targets and for Windows Forms targets. They are not implemented for Web Forms targets:

<u>Property</u>	<u>Description</u>
PaperHeight	Value for the height of the page in current units
PaperWidth	Value for the width of the page in current units
PaperOrientation	<p>Switches the values for PaperHeight and PaperWidth. Values are:</p> <ul style="list-style-type: none"> • PaperPortrait! If the current PaperWidth is larger than the current PaperHeight, switches these values so the PaperHeight is larger. If the current PaperWidth is smaller than the current PaperHeight, does not change these values. Value of 0 returned for this setting. • PaperLandscape! If the current PaperHeight is larger than the current PaperWidth, switches these values so the PaperWidth is larger. If the current PaperHeight is smaller than the current PaperWidth, does not change these values. Value of 1 returned for this setting. The PaperOrientation value is also set to this value when the PaperHeight and PaperWidth are the same size.

Changing current units By default, the current units are set to 1/1000 of an inch, so to display a paper height of 11 inches, you must set the PaperHeight value to 11000. If you enable the PopMenu property of the RichTextEdit control, you allow the application user to bring up the Rich Text Object dialog box and change the current units to 1/1000 of a centimeter. If the user switches the unit value to centimeters, the values for PaperHeight and PaperWidth are multiplied by 2.54.

By default, the values you set for PaperHeight and PaperWidth are used for printing as well as for screen display. When you set either or both of these screen display values, the default value for the Size drop-down list on the Print Specifications page of the Rich Text Object dialog box changes to Customized.

Similarly, the default value for page orientation on the Print Specifications page changes when you assign a value to the PaperOrientation property. Application users can modify the print

specifications from the Rich Text Object dialog box at runtime, but only if you set the PopMenu property of the rich text object to true.

[Specifying null values as retrieval arguments](#)

When you preview a DataWindow that takes retrieval arguments in the DataWindow painter, you may want to specify that one or more arguments has a null value. The Retrieval Arguments dialog box now has a check box next to each retrieval argument. Select the check box to specify that the argument has a null value. Typing a value in the Value box clears the check box.

[Informix 10 support](#)

PowerBuilder 11.1 introduces a new database interface for Informix databases. The I10 interface in *PBI10110.DLL* requires the Informix Client SDK 2.9 or later for Informix application development and Informix Connect 2.9 or later for runtime deployment. The new interface supports access to Unicode databases and several features that are available in recent versions of IBM Informix Dynamic Server (IDS):

- [Accessing Unicode data](#)
- [Assigning an owner to the PowerBuilder catalog tables](#)
- [Support for long object names](#)
- [Renaming an index](#)
- [SQL statement caching](#)
- [Creating and dropping indexes without locking](#)
- [Column-level encryption](#)
- [Using multiple OUT parameters in user-defined routines](#)

Restriction You cannot use the IN9 interface and the I10 interface in a single PowerBuilder session.

[Accessing Unicode data](#)

PowerBuilder can connect, save, and retrieve data in ANSI/DBCS databases using the IN9 interface, but the IN9 interface does not support Unicode databases. The Informix I10 interface supports ANSI/DBCS and Unicode databases.

The I10 native interface uses the Informix GLS (Global Language Support) API for global language support. The native interface uses three DBParms to help you set up the locale used in the current connection:

- Client_Locale
- DB_Locale
- StrByCharset

These parameters are available on the Regional Settings tab page in the Database Profile Setup dialog box.

Client_Locale

Client_Locale specifies the value of the Informix environment variable CLIENT_LOCALE. The format is language_territory.codeset. For example:

```
Client_Locale='en_us.1252'  
Client_Locale='en_us.utf8'
```

The I10 interface uses this setting to access string data in an Informix database and to process SQL statements. If you do not set the DBParm, the default locale value is based on the OS locale.

DB_Locale

DB_Locale specifies the value of the Informix environment variable DB_LOCALE. The format is language_territory.codeset. For example:

```
DB_Locale='en_us.1252'  
DB_Locale='en_us.utf8'
```

DB_LOCALE specifies the language, territory, and code set that the database server needs to correctly interpret locale-sensitive datatypes such as NChar and NVarChar in a specific database. The code set specified in DB_LOCALE determines which characters are valid in any character column, as well as in the names of database objects such as databases, tables, columns, and views. If you do not set the DBParm, the I10 interface assumes that the DB_LOCALE value is the same as the CLIENT_LOCALE value.

You can set the CLIENT_LOCALE and DB_LOCALE environment variables directly using the Informix Setnet32 utility, available in the Utilities folder for the Informix database interfaces in the Objects view in the Database painter or the Database Profiles dialog box.

For more information about the Informix CLIENT_LOCALE and DB_LOCALE environment variables, see the IBM Informix GLS User's Guide, currently available at the Informix library Web site at <http://publib.boulder.ibm.com/epubs/pdf/25122820.pdf> .

StrByCharset

The StrByCharset DBParm specifies how to convert string data between PowerBuilder Unicode strings and Informix client multibyte strings. By default, string conversion for UTF-8 code sets is based on the UTF-8 code set, and string conversion for non-UTF-8 code sets is based on the current OS code page. If StrByCharset is set to 1 (true), string conversion is based on the code set specified in the DBParm Client_Locale.

[Assigning an owner to the PowerBuilder catalog tables](#)

When you use the I10 interface, you can use the PBCatalogOwner DBParm on the System tab page to assign a nondefault owner to the extended attribute system tables. For ANSI-compliant databases, the owner name that you specify must be unique but the table name does not have to be unique. You can create multiple sets of catalog tables prefaced with different user names. However, if the database is not ANSI-compliant, the table name must be unique, so that only one set of catalog tables can be created with an assigned owner name.

[Support for long object names](#)

The I10 interface supports Informix long object names with up to 128 characters.

[Renaming an index](#)

With IDS 9.2.1 and later, you can change the name of an index in the Database painter when you are connected using the I10 interface. The I10 interface uses the IDS RENAME INDEX statement to change the name of the index. You need only drop and recreate the index if you want to make other changes.

[SQL statement caching](#)

In IDS 9.2.1 and later, the database server uses the SQL statement cache (SSC) to store SQL statements across user sessions. When any user executes a statement already stored in the SQL statement cache,

the database server does not parse and optimize the statement again, resulting in improved performance. The statement must be a **SELECT**, **UPDATE**, **DELETE**, or **INSERT** statement, and it cannot contain user-defined routines.

There are several ways to configure caching on the server. The **SET STATEMENT CACHE** statement takes precedence over the STMT_CACHE environment variable and the STMT_CACHE configuration parameter. You must *enable* the SQL statement cache, either by setting the STMT_CACHE configuration parameter or by using the Informix **onmode** utility, *before* the **SET STATEMENT CACHE** statement can execute successfully.

You can set the StmtCache DBParm on the System tab page in the Database Profile Setup dialog box for I10 connections to turn SQL statement caching on or off on the client. However, the server must be configured to support SQL statement caching before you can access the cache from the client.

For more information about Informix SQL statement caching, see the IBM Informix Dynamic Server Performance Guide at <http://publib.boulder.ibm.com/epubs/pdf/25122960.pdf> .

[Creating and dropping indexes without locking](#)

In IDS 10.0 and later, the SQL syntax of **CREATE INDEX** and **DROP INDEX** supports the **ONLINE** keyword to create or drop an index in an online environment where the database and its tables are continuously available. When you use the **ONLINE** keyword to create or drop an index, data definition language (DDL) operations execute without applying an exclusive lock on the table on which the specified index is defined.

If you use **CREATE INDEX ONLINE** to create an index on a table that other users are accessing, the index is not available until no users are updating the table. If you issue **DROP INDEX ONLINE** to drop an index, no users can reference the index, but concurrent data manipulation language (DML) operations can use the index until the operations terminate. Dropping the index is deferred until no users are using the index.

You can set the OnlineIndex static DBParm on the System tab page in the Database Profile Setup dialog box for I10 connections to specify that the Database painter should use the **ONLINE** keyword when you create or drop an index.

Clustered index not supported You cannot create a clustered index using online mode because it is not supported by IDS.

Column-level encryption

In IDS 10.0 and later, the SQL statement **SET ENCRYPTION PASSWORD** can improve the confidentiality of data and support data integrity by defining or resetting a password for encryption and decryption of data at the column level.

You can set the EncryptionPass and Hint static DBParms on the System tab page in the Database Profile Setup dialog box for I10 connections to specify a password and a hint to help you remember the password. The application uses built-in Informix functions to encrypt and decrypt character data.

Using multiple OUT parameters in user-defined routines

In a user-defined routine (UDR), an OUT parameter corresponds to a value returned through a pointer. Before IDS version 9.4, IDS supported no more than one OUT parameter in a UDR, and any OUT parameter was required to appear as the last item in the parameter list. IDS version 9.4 drops these restrictions, supporting multiple OUT parameters anywhere in the parameter list of the UDR. This feature is available when you use the I10 interface. It provides greater flexibility in defining UDRs, and removes the need to return collection variables in contexts where multiple returned values are required.

To return OUT parameters from a UDR, you must use statement local variables (SLVs). In the following statement, the OUT parameter in the UDR **myfunc** is defined using the SLV syntax *slvname#out_param_type*.

```
SELECT sales FROM mytable WHERE myfunc(10, sales#money) < 1000
```

Informix does not support invoking a UDR with OUT parameters using an **EXECUTE** statement, therefore multiple OUT parameters are not supported in PowerBuilder remote procedure calls and embedded SQL **EXECUTE PROCEDURE** commands.

Statement caching for Oracle 10g

Statement caching in Oracle 10g provides and manages a cache of statements for each session. On the server, cursors are ready to be used without the need to parse the statement again before execution.

Statement caching can be used with connection pooling and improves performance and scalability.

To set the maximum size of the statement cache, set the StatementCache DBParm to the number of statements you want to cache. The default value is 0, which means that statement caching is disabled in the current session. This DBParm is available only in the O10 database interface.

Identity columns in Adaptive Server 15

In Adaptive Server® Enterprise 15.0 and later, you can use the following datatypes as identity columns: bigint, int, numeric, smallint, tinyint, unsigned bigint, unsigned int, and unsigned smallint. This support has been added to the ASE and SYC database interfaces for PowerBuilder 11.1.