

Team Developer New Features



Trademarks

GUPTA, SQLBase, SQLRouter, Team Developer, and SQLTalk are trademarks of Unify Corporation registered in the United States and/or other countries. SQLWindows is a registered trademark of Microsoft Corporation, and is used under license to Unify Corporation, Centura and Centura Software are registered trademarks of Centura Software Corporation, and are used under license to Unify Corporation.

Team Object Manager, Web Developer, QuickObjects, Object Nationalizer, Report Builder, Connectivity Administrator, Development Kit, Component Development Kit, SQLBase Resource Manager, and SQLConsole are names of products from Unify Corporation.

Adobe is a trademark of Adobe Systems, Incorporated.

IBM, OS/2, NetBIOS, and AIX are registered trademarks of International Business Machines Corporation.

Linux is a trademark of Linus Torvalds.

Red Hat is a trademark of Red Hat, Incorporated.

Novell and Netware are trademarks of Novell, Inc.

SUSE is a trademark of SUSE LINUX AG.

Java, JavaScript, and Solaris are trademarks of Sun Microsystems, Incorporated.

Microsoft, Outlook, PowerPoint, Visual C++, Visual Studio, Internet Explorer, Internet Information Server, DOS, Win 32, Windows, ActiveX, MSDN, SQL Server, and Visual Basic are either registered trademarks or trademarks of Microsoft Corporation in the United States of America and/or other countries.

Netscape FastTrack and Navigator are trademarks of Netscape Communications Corporation.

All other product or service names mentioned herein are trademarks or registered trademarks of their respective owners.

Copyright

Copyright © 2008 by Unify Corporation. All rights reserved.
Mar 2008

1	New Features in Team Developer 5.2 SP 3	31
	"Default" Push Button Appearance.	1
2	New Features in Team Developer 5.2 SP 1	11
	Images in Popup Menus.	1
	Picture Object Improvements	1
	Multi-line ToolTips and Column Titles	1
	SalFileOpenCP.	2
	Unicode in Informix.	2
3	New Features in Team Developer Version 5.2	1
	CDK Update	1
	Compiler Tab in Preferences	1
	Connection Strings	2
	Date and DateTime Messaging	2
	Embedded Images	3
	Grid Control	3
	HTML Help	4
	Max Data Length: Character & Byte.	4
	Multiple Object Select.	4
	New Binary Data Type variable	4
	Push Button and Option Button Attributes	4
	Report Builder Text Rotation	5
	Ribbon Control Enhanced	5
	Rich Text Control	5
	Support Importing Web Services which require authentication	5
	Support for Latest Windows Versions	6
	Tab Control.	6
	Table2Grid Utility	6
	Tooltips.	7
	Warning message when using deprecated api's	7
	Watermark Image Support (Report Builder).	7

4	New Features in Team Developer Version 5.1 SP3 HotFix 3	1
	Performance enhancement for applications that are intensive with respect to creating and destroying windows.	1
5	New Features in Team Developer Version 5.1 SP3 HotFix 2	1
	Added two new Sal API's	1
	New PDF Settings dialog box in Report Builder.	1
6	New Features in Team Developer Version 5.1 SP2	1
	Better Win32 API Migration Support	1
7	New Features in Team Developer Version 5.1	1
	Unicode Support	1
	Web Services support	1
	Connectivity.	2
	Report Builder PDF support	2
	Support for Themes.	2
	Date Picker	2
	Date Time Picker.	2
	Tab Control	3
	Ribbon Bar	3
	Grid Control.	3
	WebHelp	3
8	New Features in Team Developer Version 4.2	1
	Team Object Manager Enhancements	1
	Miscellaneous Enhancements.	1
9	New Features in Team Developer Version 2005.1	1
	XML Operations	1

- UDV Serialization and Deserialization 1
- XML Sample Applications. 2
- Find-and-Replace Enhancements 2
- Report Builder Enhancements 2
- Table Window Enhancements 3
- ODBC Router Enhancements 3
- Miscellaneous Enhancements 4
- New Reserved Words in SQLWindows 4

10 New Features in Team Developer Version 2005 1

- Linux Compatibility 1
- Improved Performance. 1
- Find-and-Replace Enhancements 1
- Report Builder Enhancements 2
- Active Coding Assistant Enhancements. 3
- OLE DB Enhancements 3
- Other Enhancements 4
- New Reserved Words in SQLWindows 4

11 New Features in Team Developer Version 3.1 1

- New features in SQLWindows client application GUI . . . 1
 - Table windows extended GUI 1
 - Resizable dialog boxes and toolbars 1
- New features in SQLWindows developer IDE 2
 - Debugging and breakpoint enhancements 2
- More SQLWindows new features. 3
 - Event logging 3
 - Tracing 3
 - XML support for table windows 4
 - COM+ performance improvements 4
 - Enhanced OLE DB features. 5
 - Miscellaneous enhancements 5

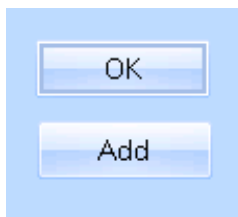
Report Builder enhancements.	6
Use of multiple paper trays	6
New formatting controls in the ribbon toolbar	6
New abilities to use conditional formatting	6
Manual or conditional page break.	7
Calculate total pages	7
International language date display	8
Enhanced Oracle router	8
Web Application Manager (WAM) for LINUX	8
SQLBase version 8.5 integration.	8
Multiple SQLBase Installations	9
SQLBase configuration file (SQL.INI)	9

Chapter 1

New Features in Team Developer 5.2 SP 3

"Default" Push Button Appearance

The `SalSetDefButton` function sets a push button as the "default" so that the user can activate the indicated button by striking the Enter key. The appearance of a "default" pushbutton has been changed in SP3 in order to fix an ambiguity issue between "selected" and "default" push buttons. The "default" push button now has an outline, as pictured below.



Chapter 2

New Features in Team Developer 5.2 SP 1

Images in Popup Menus

A "Picture File Name" attribute has been added to Popup Menus. When you select a popup menu in the heirarchal view, the "Picture:" field becomes active. Use this field to select an image to add to the popup menu.

See “Adding an image to a popup menu” on page 6-12.

Picture Object Improvements

The picture objects in Team Developer and Report builder have been improved. They now support:

- 16 & 32 bit images, including full transparency support.
- Custom TIFF tags.
- Image data including compression type, image format and other image-specific data.

Multi-line ToolTips and Column Titles

In the attribute inspector, the ToolTip and Column Title attributes can contain more than one line. For both attributes, use Ctrl-Enter to start a new line.

Note: In the ToolTip attribute, "\n" still works to start a new line, but Ctrl-Enter functionality has been added to match the Column Title attribute.

See *SQLWindows Objects* on page 5-1 for more information on each object, or *Attribute Inspector* on page 2-18 for more information on the attribute inspector.

SalFileOpenCP

This new API performs the same function as SalFileOpen, with an additional parameter to specify a code page for the file to be read. This eliminates ambiguity in foreign characters. The following code page constants have been provided:

- ENC_ANSI
- ENC_OEMCP
- ENC_MACCP
- ENC_UTF7
- ENC_UTF8

See the in-build help for details on this API.

Unicode in Informix

Team Developer now supports using unicode through the informix native router.

See “Unicode” on page 3-14.

Chapter 3

New Features in Team Developer Version 5.2

CDK Update

The CDK outline interface has been updated to cater to the new controls that were added in 5.1 and 5.2:

These controls are as follows:

- Date Picker
- Date Time Picker
- Grid
- Rich Text Control
- Tab Bar

Compiler Tab in Preferences

The dialog box invoked by selecting Preferences from the Tools menu contains a new tab entitled "Compiler." In this tab, you can choose whether the compiler displays a warning message when a deprecated function is used.

Connection Strings

A new [ConnectionStrings] section has been added to SQL.ini. This section contains connection strings in a flexible format consisting of multiple name/value pairs in this form:

```
“name1=value1;name2=value2;...”
```

The routers still support the existing sections, but in addition, the TD global variable SqlDatabase can now optionally take a connection string in the format:

```
“ConnectionString:<Actual Connection String Here>”.
```

This enables TD developers to more flexibly specify a database connection.

See *Connection Strings* on page 2-7 of the document entitled *Initializing and Testing Your Connection*.

Date and DateTime Messaging

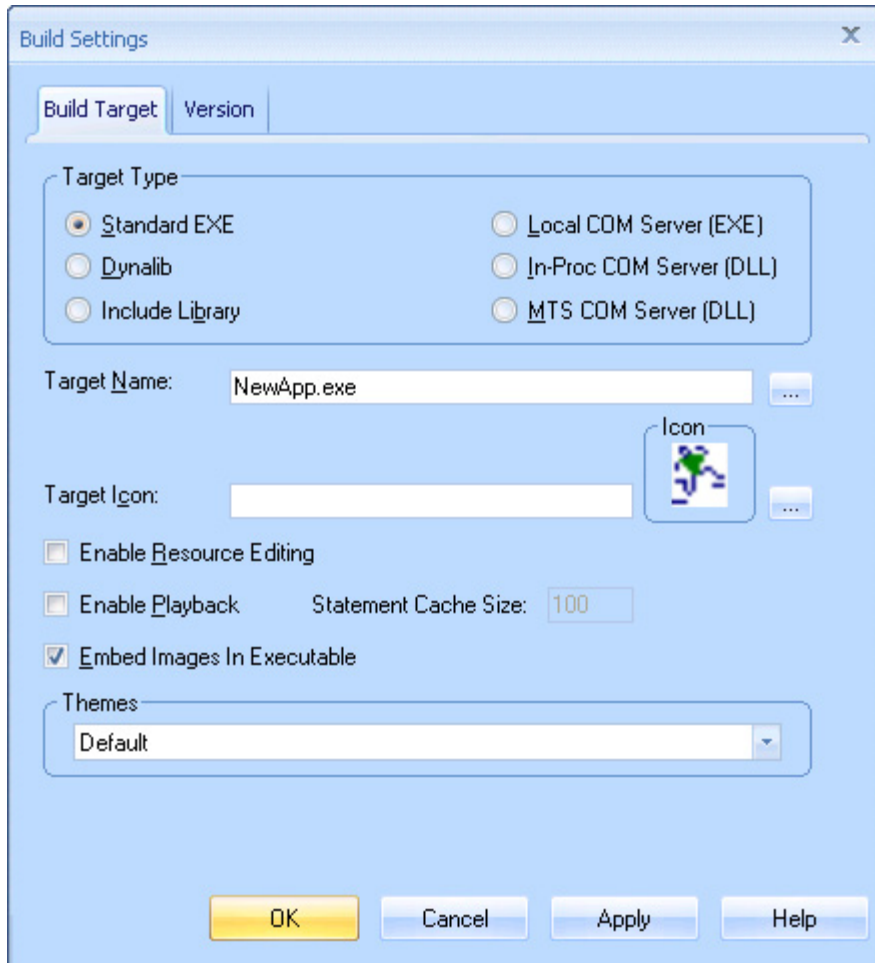
Date and DateTime Pickers now support a messaging interface.

The following messaging interfaces are supported:

- SAM_AnyEdit - To capture any keystroke
- SAM_FieldEdit - Sent when changes to a field have been done
- SAM_KillFocus - When the focus goes away from the current object
- SAM_SetFocus - When the object receives the focus
- SAM_Validate - To validate contents after editing

Embedded Images

Images are now embedded in .EXE files by default. This feature can be disabled from the Build Settings dialog box (select "Build Settings" from the Project menu and uncheck the last checkbox - see below).



Grid Control

Extensive support for table window api's. For information on the Grid Control, including sorting and grouping, see *Grid* on page 5-26 in the document *Developing with SQL Windows*.

HTML Help

WinHelp is not supported in Windows Vista without installing a WinHelp viewer, so Team Developer Help is now presented in a compiled HTML Help file (sqlwin.chm) which is supported by Vista.

Together with this new help system, a new sal api enables SQLWindows Applications to reference HTML Help files. This new api is called **SalHtmlHelp**.

Max Data Length: Character & Byte

By default, the Max Data Length attribute is measured by character length, regardless of how many bytes the data contains. This makes no difference in English strings, because 1 character requires 1 byte. However, as an example, a Japanese kanji character requires 2 bytes. This can cause ambiguity in the length of the string, because a Japanese string with a character length of 5 has a byte length of 10.

A new boolean system variable eliminates this ambiguity by allowing you to indicate whether the Max Data Length attribute is determined by character length or byte length. This clarification helps prevent database insert errors, and provides compatibility with earlier Team Developer versions.

Set **bMaxDataLengthInBytes** to TRUE On SAM_AppStartup to set the Max Data Length attribute as byte length.

Multiple Object Select

Multiple objects can be selected in the outline or in the form layout. Use Ctrl-click to select multiple objects and Shift-click to deselect one of the objects. See *Multiple Object Select* on page 2-19 in the document *Developing with SQL Windows*.

New Binary Data Type variable

Adds the Binary Data Type variable. See *Data types* on page 7-2 in the document *Developing with SQL Windows*.

Push Button and Option Button Attributes

Three attributes have been added for the push button and option button controls:

- Image Alignment
- Text Alignment
- Text Image Relation

These provide better control with respect to the look and feel of push/option buttons. See *Push button attributes* on page 5-30 in the document *Developing with SQL Windows*.

Report Builder Text Rotation

Report Builder introduces a new feature for rotating text 0, 90, 180, or 270 degrees.

Ribbon Control Enhanced

Includes new api `SalSetActiveRibbonPage` (see *SQLWindows in-build help*).

Rich Text Control

Adds Rich Text Control object, including a corresponding set of properties in the attribute inspector, a new pushbutton in the controls palette, and 29 `SalRTF` functions.

See *Rich text* on page 5-48 and *Controls palette* on page 2-26 in the document *Developing with SQL Windows*.

See also the *Rich Text Control Functions* chapter in the document *Team Developer - API Reference*.

Using Rich Text control, Team Developer applications can save full .rtf files, which are recognized by other applications that accept .rtf format, such as Microsoft Word. Additionally, RTF objects and documents can store complex data such as images, tables, hyperlinks, and unicode text.

A Rich Text Control object has also been added to Report Builder. Place a Rich Text field in a report just as you would a Text Field or Data Field.

Support Importing Web Services which require authentication

Three methods of web service authentication are now supported:

- HTTP Basic
- HTTP Digest
- WS-SECURITY

See *Authentication* on page 6-1 in the document *Developing with SQL Windows*.

Support for Latest Windows Versions

Team Developer 5.2 is supported on Windows Vista, Windows Server 2008, and Windows 7.

Tab Control

SalTabHidePage and SalTabShowPage were the only available tab api's in the 5.1 release. The 5.2 release includes the following tab api's:

1. **SalTabAddPage** - Add a page at the end or at the place where user can insert
2. **SalTabPageCount** - Returns the number of tab pages
3. **SalTabRenamePage** - Rename the tab page
4. **SalTabRemovePage** - Remove the tab page
5. **SalTabAssociateChild** - Associate a child to a tab page
6. **SalTabDisassociateChild** - Disassociate a child for a tab page
7. **SalTabSetActivePage** - Setting the active tab
8. **SalTabDisablePage** - Disable tab page
9. **SalTabEnablePage** - Enable tab page
10. **SalTabGetPageTitle** - Returns the tab page name depends on the index passed
11. **SalTabGetPageIndex** - Returns the tab index depends on the name passed
12. **SalTabGetActivePage** - Returns the current active tab page

Table2Grid Utility

The Table2Grid utility has been provided to migrate existing applications' table windows to grid controls. To run the utility, double-click on **Table2Grid.exe** in your root Team Developer installation directory.

Note: The Table2Grid utility can also be run from the command line as follows:

```
Table2Grid "TableApplication1.app" "c:\temp"
```

Parameter 1 is a string which specifies the app, apt, or apl which has the tables.

Parameter 2 is a string which specifies an output directory to store the converted file.

Tooltips

A ToolTip property has been added to many of the UI controls together with sal api's SalSetTooltip and SalGetTooltip. See in-build help for details on these sal api's.

Warning message when using deprecated api's

Displays a warning message when using an obsolete api. This feature can be disabled through the Tools menu (Tools / Preferences / Compiler Tab).

Watermark Image Support (Report Builder)

Allows you to use images or text as watermarks, as well as set watermark properties (size, transparency, diagonal/horizontal layout, etc). See *Watermarks* in the document *Report Builder Workspace*.

Chapter 4

New Features in Team Developer Version 5.1 SP3 HotFix 3

Performance enhancement for applications that are intensive with respect to creating and destroying windows.

Code was added to check if a window has destroyed itself during creation as the create window process is finishing. If this condition arises we move destruction of this window from the end of the message queue to the front of the message queue and return a NULL handle.

This behaviour can be over-ridden thereby reverting back to the same behaviour prior to TD 5.1 SP3 HF3 by adding a new DWORD value called

DeferWindowSelfDestruct with a non zero value to the registry key:

HKEY_CURRENT_USER\Software\Gupta\SQLWindows 5.1\Settings

Chapter 5

New Features in Team Developer Version 5.1 SP3 HotFix 2

Added two new Sal API's

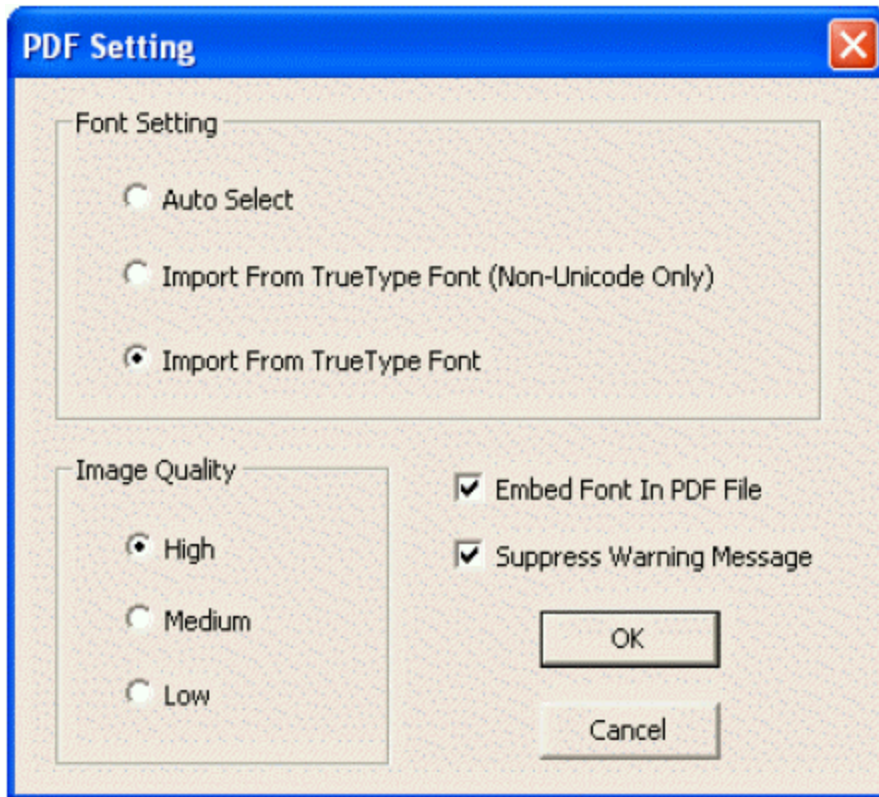
The following new sal api's were added:

- SalReportPrintToFileEx
- SalPrtGetPrinterTrays

New PDF Settings dialog box in Report Builder.

When you want to save a report to a PDF file you will now have a PDF Settings dialog box appear as shown below.

These settings are exposed in the new sal api SalReportPrintToFileEx. If you look at the last 4 parameters of this api you will understand its operation.



Chapter 6

New Features in Team Developer Version 5.1 SP2

Better Win32 API Migration Support

With this release of Team Developer we have identified the common win32 api migration issues and have put in place various infrastructure support to aid with this migration. See "Developing with SQLWindows": *Chapter 23, Team Developer Win32 API Migration*.

Chapter 7

New Features in Team Developer Version 5.1

Unicode Support

Unicode support is now available in SQLWindows, Report Builder, CDK, Quick Objects, Database Explorer, Web App Manager, Visual ToolChest, Object Nationalizer and Team Object Manager. Also all applications created before Team Developer 5.1 will be unicode enabled when opened in SQLWindows 5.1.

If you want to minimize the impact of migrating your dlls which were written against previous versions of Team Developer (i.e non-unicode versions) you need to read the document titled 'Team Developer ASCII API' which outlines how to do this. This document together with an example is available when you install the Team Developer Samples. It is located at: `.../Samples/SQLWindows/SalASCII/`

Web Services support

A new wizard has been added to the SQLWindows tools menu which allows you to consume web services. Using the Web Services Wizard, one can generate a SAL interface library containing all the logic needed to interact with a remote web service. Using the popular and stable Axis2 API will allow you to communicate using either the RPC or document-style messaging, and includes support for complex types, arrays, lists, nested types, default values, type inheritance, any elements, and imports.

For more details about Team Developer web services click on the following link:
[Team Developer Web Services](#)

There is also some sample Web Services available if you install the Team Developer samples.

Note: You need to have .NET 2.0 on your machine to use the new Web Services Wizard.

Connectivity

- Updated the routers to support the latest client drivers for Informix, Microsoft SQL Server, Oracle, SQLBase, and Sybase.
- Unicode support has been added for Microsoft SQL Server, Oracle, SQLBase, and Sybase routers.
- Support for the SQL Native Client driver for Microsoft SQL Server 2005 as been added.

Report Builder PDF support

Added the ability to save a report as a PDF document. The SAL api `SalReportPrintToFile` has been updated to support the PDF format.

Support for Themes

Built-in UI themes allow your application to have the overall look and feel of popular Microsoft products such as Microsoft Office 2007/2003/XP/2000 and Microsoft Visual Studio 2005.

At build time you can choose one of these themes or you can programmatically set the theme at runtime.

Date Picker

The new date picker control looks like and has similar features as the mini-calendar in Microsoft Outlook.

Supports two SAL api's, `SalSetDateTime` & `SalGetDateTime`.

Date Time Picker

The date and time picker control provides a simple and intuitive way to exchange date and time information with the user. The date and time picker control supports 12 and 24 hour time formats and can be customized to show only date, only time, and

both date and time.

The embedded drop-down button invokes a pop-up date picker which allows you to select the desired date.

Supports two SAL api's, SalSetDateTime & SalGetDateTime.

Tab Control

The new tab bar control differs from the existing Visual ToolChest (i.e. VT) tab control in that the tab bar control is not a container object like the VT tab control. Each tab bar control comprises of zero or more tab pages. Also each page can have an icon associated with it and maintains a list of controls for that page. This association is done using the Tab Child item under the tab page item. This association can be done either using the outline editor or through the layout editor

For more details about this control go to 'Developing with SQLWindows': *Chapter 5, Tab Bar*

Ribbon Bar

The ribbon bar can be used as another option for display of menu's. This feature is turned off by default. In order to enable this feature set the system variable, bMenuBarAsRibbonBar to TRUE in the 'On SAM_AppStartup' message of the application. For more details about the ribbon bar go to 'Developing with SQLWindows': *Chapter 6, Ribbon Bar*

Grid Control

A new grid control as been added which supports a subset of the Sal Table Window api's. This new control can be added to a form window or created via the outline editor. The grid also supports using a 'Column Class' in the Grid Class (i.e. called 'Child Grid Class') or Control.

For more details about this control go to 'Developing with SQLWindows': *Chapter 5, Grid*

WebHelp

Vista no longer supports winhelp format and so we moved our winhelp documentation to webhelp format.

Chapter 8

New Features in Team Developer Version 4.2

Team Object Manager Enhancements

- You can now add a file of the same name to more than one folder in a project.
- Your changes to a project's log properties are now retained.
- When checking in a file that has not been modified, the status message describing this condition now shows all text, not just the first 255 characters.
- In the Diff tool window the "thumb" on the horizontal scroll bar now performs scrolling.
- In several dialogs and menu items, keyboard accelerators have been added, corrected, or reorganized.
- Many other minor defects in Team Object Manager were corrected.

Miscellaneous Enhancements

- Performance when accessing Oracle 9i via OLEDB has been improved.
- COM servers created in Team Developer 4.2 can be silently registered and unregistered using the following syntax:
MyCOMServer.exe /RegServer /Silent
MyCOMServer.exe /UnregServer /Silent

Note: Normally no dialog will appear, but if there is an error during the process, an error dialog will appear.

- A new class definition, HTTPRequest, is added to the existing XML-related classes in library file xmllib.apl
This new class enables you to send and receive text strings, or XML documents, using the HTTP protocol.
Exchanging XML documents via HTTP is one of the principal ways businesses use SOAP, Simple Object Access Protocol, so this class will be very useful for many customers.
This version ships with sample applications that illustrate how the HTTPRequest class is used. They are not part of the regular Team Developer Samples Installer task, but can be individually uncompressed from file HttpRequest_Samples.zip.

Note: Note that your internet connection may not currently support web services, so if you wish to develop SOAP or other web service clients, you may need to configure your connection or proxy server to allow this.

Chapter 9

New Features in Team Developer Version 2005.1

XML Operations

SQLWindows applications can read, write, and manipulate XML documents using the Document Object Model. Details of these XML capabilities are available in a new chapter "Handling XML", in the book "Developing with SQLWindows" (dev.pdf). Similar information is available in online help, under the table-of-contents entry SQLWindows Programming / Handling XML. Numerous online help topics are indexed on "XML".

UDV Serialization and Deserialization

Closely related to XML support is the ability to serialize the instance variables of a UDV (user-defined variable), or array, to an XML document, and to deserialize an XML document into a UDV or array. This ability will be extremely useful, particularly to those developers who wish to save and reuse the "state" of an application. To understand how these features work, see the SQLWindows Function Reference Manual, or the online help, for functions SalXMLSerializeUDV, SalXMLDeserializeUDV, and SalXMLGetLastError. To support serialization, there are three new properties available when you modify an instance variable in a class

definition (either through Coding Assistant or by choosing "Add Next Level" in the right-click context menu):

- XML Serialize: allows you to indicate whether this specific instance variable should take part in serialization operations. Default is Yes.
- XML datatype: allows to designate a general SQLWindows datatype such as "Number" as a more specific XML datatype such as "Currency". No default value.
- XML Nulls Allowed: allows you to indicate whether null values in the XML document are permitted when deserializing XML into a UDV. Default is Yes.

XML Sample Applications

There are new sample applications to demonstrate some of the XML-related features in Team Developer 2005.1. `xml_editor_demo.apt` shows how to manipulate the XML Document Object Model. `xml_serialization_demo.apt` shows how to use serialization and deserialization with UDVs and arrays. `CreateXMLData.app` writes the result sets of SQL queries to XML, and `LoadXMLData.app` loads an existing XML document into the XML Document Object Model, then "walks" the document tree. Each of these sample applications has an accompanying documentation file with the same name and extension ".htm" (for example, `xml_editor_demo.htm`.) This documentation provides deeper insight into the sample applications.

Find-and-Replace Enhancements

Find and Replace can now be accessed from the SQLWindows toolbar. A drop-down combo box holds recently used search strings, or you can type your own. There are buttons for Find, Find Next, Find Previous, and Help. Find, Find Again, and Replace are now also context menu choices at many places in the SQLWindows outline.

Report Builder Enhancements

- The Conditional Output feature that is available in many blocks of a report is now also available for the Page Header and Page Footer blocks.
- The Define Query dialog now shows the datatypes of columns that are available for use in the query.
- The Define Query dialog now allows you to edit the names of specific columns used in the Conditions tab (Column/Formula), the Sort tab (Sort by, Asc/Desc), and the Group Conditions tab (Column/Formula).
- The Define Query dialog now permits you to enter case-sensitive database table names when delimited by quotes. Because of this new feature, you must

now take extra care when working with table names that contain one or more quotes as part of the name (some database vendors permit quote characters in the names of tables.) If you encounter such a quote character, you must "escape" it by appending a second quote character next to the one that is part of the table name. Otherwise the quote in the table name might be misinterpreted as the delimiter of a case-sensitive name, instead of part of the name itself.

- The Define Query dialog now permits you to control the sequence of columns in the result set by dragging columns to new positions using the mouse.
- LOB datatypes (BLOB and CLOB) can be passed to an input item, and Report Builder will show a picture (if the BLOB is a known graphic type such as JPG) or a text box (if it is a CLOB containing text).
- The Format Block command now provides the ability to specify general formatting properties (font, color, etc.) for all fields within a report block using a single dialog. It is still possible to override those properties for an individual field.
- The "Between Lines" setting in the Borders page of Fields properties is now saved with the rest of the report design
- It is now possible to cut and paste report elements between multiple Report Builder windows within the same Report Builder instance.
- The Formula Editor dialog now gives you the choice of displaying all formulas, used formulas, or unused formulas. Thus you can delete all unused formulas by displaying a list of them, then deleting them.

Note: The internal structure of Report Builder templates has changed. Templates that are saved in this version of Team Developer can't be opened by earlier versions.

Table Window Enhancements

New SAL function `SalTblCreateColumnEx` allows you to create new table window columns dynamically at runtime, specifying the data type of the column as well as other features such as title, display width, column position, and data length in characters.

ODBC Router Enhancements

The ODBC Router for client connectivity in the Linux version is now a native Linux component.

Miscellaneous Enhancements

For connections to Infomix databases, using function `SqlGetParameter` with parameter `DBB_BRAND` will now return value `DBV_BRAND_INFORMIXONLINE` rather than the more generic value 41.

New Reserved Words in SQLWindows

There are a large number of new function and constant names related to XML processing. However, these are stored not in the core outline of SQLWindows, but in a supplied library, `xml.lib.apl`, which you may optionally include in your application. Examine that file to see the new names. In addition to those names, the following names are reserved:

- `SalTblCreateColumnEx`
- `SalXMLSerializeUDV`
- `SalXMLDeserializeUDV`
- `SalXMLGetLastError`

Chapter 10

New Features in Team Developer Version 2005

Linux Compatibility

Team Developer now runs on Linux machines. See the release notes for specific operating systems supported. SQLWindows applications can be developed on Linux and deployed on Linux. SQLWindows source code is portable between Linux and Windows, but the compiled applications are not portable.

Improved Performance

Applications now compile much faster than in earlier versions of Team Developer. Optimization of some of the string-handling functions improve overall runtime performance by 10 to 15 percent.

Find-and-Replace Enhancements

The Find dialog box has been redesigned with two tabs, one for Find and one for Replace. Several new elements have been added to the dialog.

- Scope - specify whether entire outline or a current selection is to be searched
- Direction - specify to start search forward or backward from current position
- Find Whole Words Only - prefixed-suffixed by blank or special delimiters

- Exclude Comments
- Exclude Include Libraries
- Wrap Search (continue searching after the top or bottom of the outline is reached)

If you have highlighted text within the outline and you then invoke the Find/Replace dialog, that text will be used to initialize the "Find What" datafield in that dialog.

Report Builder Enhancements

- Menu item Report, Report Information allows you to enter a comment for a report template or CQT file. New function `SalReportGetFileInfo` retrieves that comment. In addition, that comment will appear, if present, in the File/Open dialog box of Report Builder. This particular dialog does not contain the "place bar" at the left edge that was introduced in file dialogs of Team Developer 3.1, and is otherwise found throughout the file dialogs of 4.0.
- New function `SalReportSetPrinterSettings` allows you to control printer name, paper type, and page orientation at runtime. You can choose a custom paper size and specify height and width.
- New function `SalReportGetPrinterSettings` retrieves the current printer name, page width, page height, paper type, and page orientation.
- New function `SalReportSetPrinterTray` sets the input tray for a page that you specify. New parameter `RPT_StartPagePrint` has been added to the `SAM_ReportNotify` `IParam` values, for detecting the beginning of a page.
- New function `SalReportGetPrinterTray` gets the input tray that will be used for a page that you specify.
- New function `SalReportResetPages` lets you choose a break group to reset current page number to one, and reset total pages to the number of pages in that specific break group value. You can also specify this at design time in the General tab of the Format Report dialog box, in section "Reset Page Numbers".
- A new checkbox in the General tab of the Format Report dialog box allows you to use international number formatting (based on the locale language of the workstation.)
- Report Builder will allow you to change the database name saved in a QQT file when converting to a CQT file.
- When the Report Preferences dialog specifies centimeters rather than inches for the measurement units, the dimensions of pictures placed in the report will also be in centimeters.

Important note: The internal structure of Report Builder templates has changed. Templates that are saved in this version of Team Developer can't be opened by earlier versions.

Active Coding Assistant Enhancements

Active Coding Assistant (ACA) now recognizes the **New** keyword for creating objects and responds with a list of all functional classes.

When the return datatype of a function is an object, typing that function name plus a period now causes ACA to present a list of all members of that object's class.

When you are in the Outline tab of SQLWindows and you have highlighted a symbol name (such as a function name or class name), pressing F2 will "jump" to the section of the outline where the symbol is declared, so that you can view details of the implementation. Pressing Ctrl+F2 will "jump" back to your starting point.

OLE DB Enhancements

Many aspects of OLE DB connecting and processing run faster in this version than they did in prior versions.

Team Developer 2005 ships without native database routers. GUPTA recommends that developers use the OLE DB data providers for each of the supported databases. The new system variable SqlUDL, introduced in version 3.1, makes it easy to modify existing applications to use OLE DB data providers instead of native routers. SqlUDL has new behavior in version 2005. The SqlConnection function, when it finds a non-null value in SqlUDL, will use the SqlUDL information to form a connection string. But if there are any values in variables SqlDatabase, SqlUser, or SqlPassword, those values will overwrite the ones already present in the connection string. This behavior is designed to improve security by eliminating the need to store sensitive information in external UDL files.

New code block "When SessionError" allows you to supply custom error processing for OLE DB errors.

The Oracle datatype REF CURSOR is supported for Oracle OLE DB connections. In addition, PLSQL procedures that return values and/or use output parameters are also supported. To take advantage of this feature you can revise the extended properties in the UDL string to set the PLSQLRSet property to TRUE (for all connections), or you can call SqlSetParameter (ORAPROP_PLSQLRSet, TRUE) at runtime (for a single connection). GUPTA recommends that you use the Oracle OLE DB provider, not the Microsoft OLE DB provider, if you plan to use REF CURSOR.

Isolation levels RL and RO (Release Locks and Read-Only) now map to READ COMMITTED when used against other vendors' OLE DB databases. In the previous

version of Team Developer they mapped to READ UNCOMMITTED. Furthermore, the default isolation level for OLE DB connections created with function `SqlCreateConnection` is now RL (previously it was RR.)

Other Enhancements

- `SQLWindows` optionally makes automatic backup copies of applications that are being saved.
- The behavior of non-editable data fields, with regard to tab stops and selection by mouse cursor, is now a configuration option.
- The mouse wheel can now be used to scroll in table windows, edit windows, and drop-down lists.
- New function `SalEventLogSetName` allows you to specify your own source name for event log entries.

New Reserved Words in SQLWindows

The following section lists global names of functions, constants and variables related to new features in `SQLWindows`. Be sure that your existing applications do not reference these names. Names listed below have explicit index entries associated with those names in Team Developer online help, unless otherwise noted.

<code>SalReportGetFileInfo</code>	<code>SalReportGetPrinterSettings</code>	<code>SalReportResetPages</code>
<code>SalReportSetPrinterSettings</code>	<code>SalReportSetPrinterTray</code>	<code>SalEventLogSetName</code>
<code>RPT_PaperLetter</code>	<code>RPT_PaperTabloid</code>	<code>RPT_PaperA3</code>
<code>RPT_PaperA4</code>	<code>RPT_PaperA5</code>	<code>RPT_PaperB5</code>
<code>RPT_PaperCustom</code>	<code>RPT_Portrait</code>	<code>RPT_Landscape</code>
<code>ORAPROP_PLSQLRSet</code>	<code>RPT_StartPagePrint</code>	

Chapter 11

New Features in Team Developer Version 3.1

New features in SQLWindows client application GUI

Table windows extended GUI

A large number of new SAL functions permit you to set and get the following attributes at runtime:

- Font styles for an entire table window, a row, a row header, a column, a column header, or a cell.
- Foreground and background colors for an entire table window, a row, a row header, a column, a column header, or a cell.
- Separator line styles for a row or a column.
- Bitmap images for a cell, row header, or column header.

All the function names begin with **VisTbl** and are documented in SQLWindows online help under topic "Table window extended GUI functions".

Resizable dialog boxes and toolbars

Dialog boxes have three new attributes:

- Resizable
- Vertical Scroll - only enabled when the dialog is dockable or resizable

- Horizontal Scroll - only enabled when the dialog is dockable or resizable.

Toolbars have one new attribute: Resizable. It is only enabled when the toolbar is dockable.

These objects, when flagged as Resizable, present a standard "gripper" cursor to allow the user to drag a border of the object to resize it.

These objects are stateful - if a dialog is free-floating, and you resize it, then dock it, then undock it, it will "remember" the custom size that you gave it when it was last free-floating, rather than retaining its docked size or its design-time size.

New messages give you some control over sizing events:

SAM_DockResize is sent just before a resizing or a change in docking state. You can influence this event by returning a numeric value to this message. The value would contain your desired size for the object. However, in docking situations, there may be competing requests for space from other objects that are located inside the same dock bar as the object you wish to resize. For example, it's common for two or more toolbars to share the same dock bar. There may also be other dock bars active in the parent window, with other objects inside them. There is a process of negotiation that occurs when there are competing requests, and negotiation does not guarantee that the size you requested will be granted.

It is unsafe to call certain functions involving window size or location from within the context of **SAM_DockResize**. Read [SQLWindows online help](#) or *Developing with SQLWindows* for more details.

SAM_DockResizeNotify is sent after a sizing or docking state event has completed. This is your opportunity to rearrange the child windows within your object to accommodate the new size.

New features in SQLWindows developer IDE

Debugging and breakpoint enhancements

A new breakpoint management dialog box shows all breakpoints that are currently flagged in the application. A checkbox permits you to enable/disable the breakpoints from within the dialog.

In addition to breakpoints on a line of code, as in previous versions of SQLWindows, the breakpoint management dialog also permits you to enter data expressions. When the value of such an expression changes, execution pauses and a message box informs you of the change.

The breakpoint management dialog contains a "number of iterations" field that you can use to indicate that you want the breakpoint to be suppressed until that number of iterations has occurred, then break.

For a breakpoint anchored to a line of code, you can also specify a condition (a data expression) to determine whether or not the breakpoint will actually break.

The Preferences dialog previously allowed you to set a different outline text color for lines with breakpoints. Now there are two colors available, one for enabled breakpoints and one for disabled breakpoints.

When a breakpoint occurs, you can use the Step Over and Step Into operations that were available in previous versions of Team Developer. You can also use two new operations. Step Out executes until control passes to the outline level higher than the current breakpoint. Run to Cursor executes until it reaches the line of code that you have highlighted, then breaks again.

More SQLWindows new features

Event logging

Event logging allows an application to automatically log an event and optionally continue running, rather than displaying a runtime message box that needs to be answered through human intervention. Events such as SQL errors, array index errors, etc., can now be logged. This is particularly advantageous for applications that run unattended, such as COM servers.

To activate logging, call function **SalUseEventLog**. See SQLWindows online help or the Function Reference manual for detailed syntax information. Output goes to the Windows event log, or (for Window 98 and ME, which don't support event logs) to a file designated in a registry entry.

The "continue" option in SalUseEventLog behaves as follows:

Message box button choices	Behaves as if
Yes / No	You clicked Yes
Abort / Retry / Ignore	You clicked Abort
OK / Cancel	You clicked OK

Tracing

With tracing, you can direct detailed diagnostic information to several different output locations:

- The Windows event log (for Windows 98 and ME, a data file is used since the event log isn't supported by these versions.)
- A named file.

- The SQLWindows output window normally used for displaying compiler errors. Note that this option is only available when the application is running in Debug mode.
- Directly to the "stdout" window, so that trace output can be integrated with third-party debugging tools.

For more information, see online help and books for functions `SalStartTrace`, `SalEndTrace`, and `SalTrace`.

XML support for table windows

New functions in SQLWindows permit you to:

- Write out the full or filtered contents of a table to an XML document and/or schema.
- Read back an XML document and schema into a table window.

New functions include

Function	Description
<code>SalTblWriteXMLandSchema</code>	Write table window information (all rows) to an XML document and/or and XML schema.
<code>SalTblWriteXMLandSchemaEx</code>	Write table window information (selected rows) to an XML document and/or and XML schema.
<code>SalTblSetFromXMLSchema</code>	Validates that an XML schema matches a table window column layout
<code>SalTblPopulateFromXML</code>	Reads data from an XML document into the cells of a table window

New messages include:

Message	Description
<code>SAM_WriteXMLRow</code>	Sent just before a row is written to an XML file.

See SQLWindows online help under the index entry "XML" for a list of functions, constants, and messages. Also see the XML Support section in Chapter 15, *Table Windows*, in the book *Developing with SQLWindows*.

COM+ performance improvements

New function `SalComCleanupMode` allows you to choose when to release resources allocated to COM server objects:

As soon as the last object on a specific thread is destroyed (early cleanup, current behavior).

When the thread itself is destroyed (late cleanup, new alternative).

By waiting until the thread itself is destroyed, you can avoid the time expense of initializing and destroying the resources repeatedly. The time savings can be very significant in a COM server application that is called repeatedly by a client.

The default is early cleanup, since that is the method used in earlier versions of SQLWindows.

There have also been several internal improvements in object creation, function invocation, and object initialization and allocation. Cumulatively these improvements add up to noticeably quicker performance.

Enhanced OLE DB features

SqlUDL is a system variable that can contain the name of a UDL file to use for OLE DB connection information. This variable was introduced in version 3.1. One of its purposes is to ease the migration of existing SQLWindows applications from use of native routers to use of OLE DB.

To make this easy, function **SQLConnect** has been altered in SQLWindows version 3.1. **SQLConnect** now looks first at variable **SqlUDL** and, if it finds a file name in that variable, reads connection information from that file. If it finds a provider name in **SqlUDL**, it uses the provider name. If the database name or user name or password was not specified, **SQLConnect** will obtain the needed value from the values of variables **SqlDatabase**, **SqlUser** or **SqlPassword**. It forms a connection string, then makes an OLE DB connection with that string. If **SqlUDL** is null, **SQLConnect** uses the older (API and routers) method of connecting with the values of **SqlDatabase**, **SqlUser**, and **SqlPassword**. So, in many cases, existing apps simply need a few lines to set the value of **SqlUDP** and the rest of the app will run smoothly against OLE DB.

New message **SAM_SessionError** has been added to make it easy to know when a SQL error is originating from an ordinary connection or from a session. Ordinary connections will continue to use **SAM_SqlError**.

New function **SqlGetLastStatement** shows the text of the last statement executed. It is global, not dependent on cursor, so be aware of the timing of commands when using this function.

Miscellaneous enhancements

Previously undocumented SQLWindows functions **SalGetWindowLabel**, **SalPause**, and **SqlGetCursor** have now been documented.

The menu item Edit, **Replace** now has an accelerator key, **Ctrl+R**.

Report Builder enhancements

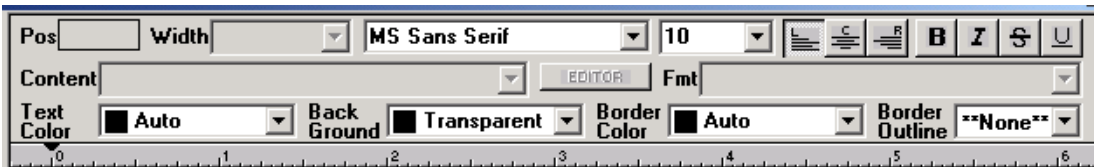
Use of multiple paper trays

On the dialog invoked by menu item Report, Format, Report, there is a new tab named Paper Source. On this tab you can choose a paper tray for the first page of the report and a different tray for all other pages. These preferences automatically adjust for the printer selected. If you change printers and the selected tray is not available on the new printer, the default tray will be used instead.

You get a second opportunity to choose trays when you invoke the Print dialog. Changes made here will not persist, but changes made in the Report/Format/Report dialog do persist.

New formatting controls in the ribbon toolbar

New controls at the bottom edge of the ribbon toolbar allow you to choose text color, background color, border color and border outline for report objects without invoking their property dialogs.

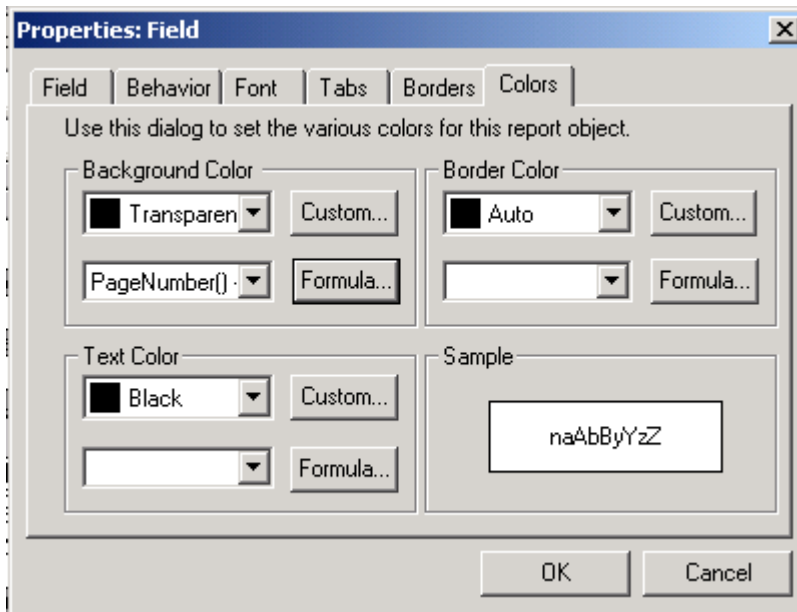


New abilities to use conditional formatting

Some elements of a report have been enhanced to include a control for selecting a formula (conditional expression). Properties that are subject to conditional formatting include:

- Text color (default: black)
- Background color (default: transparent)
- Border color (default: black)
- Font (default: Times New Roman)

- Font style (default: Regular)



You are not required to enter a formula at all when setting these properties. If you do not enter one, your chosen property value will always be used. But if you do enter a formula, the formula is evaluated at runtime. If it is TRUE (non-zero for numeric formulas, non-null for strings) then your selected property value is used. If it is FALSE, the default value of the property is used.

All five of the properties above can be conditionally formatted for fields. For lines, only background color and border color can be conditionally formatted.

Manual or conditional page break

In the Behavior tab of the Line properties dialog, there is a new option to choose an expression or formula to be associated with a page break. If the formula evaluates to TRUE, the page break will occur; if it is FALSE, the page break will not occur. It is still possible to specify a page break that is unconditional (always breaks).

Calculate total pages

New function **TotalPages()** can be invoked to determine the total number of pages in the report, even at the beginning of printing. This is useful for creating "Page 1 of 15" style strings for headers and footers.

International language date display

A new option for menu item Report, Format, Report allows you to click a checkbox titled "Enable Intl. Date Display". If you do so, function `CurrentDate()` will return a date string formatted in the workstation's locale language. In addition, new function **`DateToStrPictureIntl()`** will return a string formatted in the workstation's locale language, regardless of the setting of the new checkbox.

Enhanced Oracle router

The router has been upgraded from use of Oracle OCI 7 APIs to optional use of OCI 8 APIs. This permits you to use some of the Large Object (LOB) datatypes available in OCI 8. Previous versions of the router used the Long Raw datatype, but since Oracle has deprecated this datatype beginning with OCI version 10, it is important that you have enough time to move your datatypes to LOB while still being able to run applications that use Gupta routers.

The router determines which APIs to use based on a new SQL.INI keyword. **`USELOB`** has a default value of 0 (older APIs will be used). You can also set its value to 1 (OCI 8 APIs will be used). LOB datatypes work only when `USELOB=1`.

Another way of getting and setting this option is through the use of new parameter **`DBP_ORAUSELOB`** with the `SqlGetParameter` and `SqlSetParameter` functions.

LOB datatypes supported include CLOB and BLOB.

Web Application Manager (WAM) for LINUX

Some of the WAM components have been adapted for use on Linux servers running the Apache web server 1.3.x or 2.x. The CGI and DSO components now reside there, along with a new component, the Gupta Naming Services daemon. This means that Team Developer applications can now use Apache web servers as well as Microsoft web servers. For detailed information on installation and configuration of these components, see the chapter *WAM for Linux* in *Building Web Applications with Gupta*.

SQLBase version 8.5 integration

Team Developer 3.1 ships with SQLBase version 8.5. There are many major new features in 8.5, all of which are described in the book *SQLBase Guide to New Features*. In this section we will only discuss the features that are of greatest interest to client application developers using Gupta SQLBase or Gupta routers.

Multiple SQLBase Installations

SQLBase now has the ability to support more than one installation of SQLBase on a computer. Multiple instances of the SQLBase database engine, even different versions of SQLBase, can run simultaneously. Multiple client configurations can also run simultaneously without interfering with each other.

The configuration differences that make multiple installation possible are only available in SQLBase 8.5. You can run one or more SQLBase 8.5 database engines concurrently with one earlier version of SQLBase, but you cannot run multiple earlier versions simultaneously.

SQLBase configuration file (SQL.INI)

In order to support multiple installations, the configuration file, always named SQL.INI in versions prior to 8.5, now has a flexible name and path specification. You may use whatever name you like in place of SQL.INI. *We will continue to use the name SQL.INI throughout SQLBase documentation*, although your actual file name may be different.

Client applications written with Team Developer also use the configuration file to determine what database servers and communication protocols are available. With the possibility of multiple configuration files and multiple servers active on a single machine, there is a need for the client application to indicate which configuration it wants to use. This need exists both at design time and at run time.

The selection of a configuration file at design time is handled by a new option in the Preferences dialog of SQLWindows. The General tab of that dialog contains a control that allows you to type or browse a specific filename. This choice is used by SQLWindows and by other Team Developer design tools, such as SQLTalk, Report Builder, and Team Object Manager.

These tools can also accept a command line argument specifying what configuration file to use. See the book for each tool for exact syntax descriptions for the argument.

SQLWindows also provides a means of specifying a configuration file at runtime. An application can supply a simple or fully qualified filename to new system variable **SqlINI**, and that configuration file will be used when making database connections.

A simple filename, without path, causes SQLWindows to search for that filename in the current path of the application. A fully qualified file name causes SQLWindows to search only in that specified directory.

If you do not make an explicit configuration file choice via the Preferences dialog or the SqlINI variable, Team Developer will use the method of locating SQL.INI that was used in previous versions.

Variable `SqlINI`, once changed, affects all future connections, through both `SqlConnection` and `SqlCreateSession`. If a connection is already open when `SqlINI` is changed, the next call that references that connection will return an error. It is strongly recommended that open connections be closed before changing the value of `SqlINI`.

The value of the current configuration file can be retrieved through parameter `DBP_SQLCONFIGFILENAME` in conjunction with function `SqlGetParameter`. This works whether `SqlINI` has a non-null value or not.

A

Apache web server 8

B

breakpoint management dialog 2

breakpoints on data expressions 2

C

CGI 8

COM+ performance 4

CurrentDate 8

D

DateToStrPictureIntl 8

DBP_ORAUSELOB 8

DBP_SQLCONFIGFILENAME 10

Debugging 2

DSO 8

E

Event logging 3

F

formatting controls 6

H

Horizontal Scroll 2

L

Large Object (LOB) datatypes 8

LINUX 8

M

multiple paper trays 6

N

negotiation during resizing 2

number of iterations in breakpoints 2

O

OLE DB 5

Oracle OCI 8 8

R

Report Builder 6

 Calculate total pages 7

 conditional formatting 6

- conditional page break 7
- formatting controls 6
- International language date display 8
- Resizable dialog boxes and toolbars 1
- Resizable property 1
- Run to Cursor, debugging 3
- S
- SalComCleanupMode 4
- SalGetWindowLabel 5
- SalPause 5
- SalTblPopulateFromXML 4
- SalTblSetFromXMLSchema 4
- SalTblWriteXMLandSchema 4
- SalTblWriteXMLandSchemaEx 4
- SalUseEventLog 3
- SAM_DockResize 2
- SAM_DockResizeNotify 2
- SAM_SessionError 5
- SAM_WriteXMLRow 4
- SQL.INI 9
- SQLBase 8
- SQLConnect 5
- SqlConnect 10
- SqlCreateSession 10
- SqlGetCursor 5
- SqlGetLastStatement 5
- SqlGetParameter 10
- SqlINI 9
- SqlUDL 5
- Step Out, debugging 3
- T
- Tab Control in version 5.2 6
- Table windows extended GUI 1
- TotalPages 7
- Tracing 3

U

USELOB 8

V

Vertical Scroll 1

VisTbl* 1

W

Web Application Manager 8

X

XML 4