

**TEAM DEVELOPER™**

# TEAM DEVELOPER™

New & Changed Features

Product Version 6.2.3



Team Developer™: New & Changed Features, Product Version 6.2.3

The software described in this book is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Last updated: December 10, 2014.

## Legal Notice

Copyright © 2014-2015 Gupta Technologies, Inc. All rights reserved.

Gupta, Gupta Technologies, the Gupta logo, Gupta Powered, the Gupta Powered logo, ACCELL, Centura, Centura Ranger, the Centura logo, Centura Web Developer, Component Development Kit, Connectivity Administrator, DataServer, DBIntegrator, Development Kit, eWave, Fast Facts, NXJ, Object Nationalizer, Quest, Quest/Web, QuickObjects, RDM, Report Builder, RPT Report Writer, RPT/Web, SQL/API, SQLBase, SQLBase Exchange, SQLBase Resource Manager, SQLConsole, SQLGateway, SQLHost, SQLNetwork, SQLRouter, SQLTalk, Team Developer, Team Object Manager, TD Mobile, Velocis, VISION, Web Developer and WebNow! are trademarks of Gupta Technologies and may be registered in the United States of America and/or other countries. SQLWindows is a registered trademark and TeamWindows, ReportWindows and EditWindows are trademarks exclusively used and licensed by Gupta Technologies.

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation/reverse engineering. No part of this document may be reproduced in any form by any means without prior written authorization of Gupta Technologies Corporation and its licensors, if any.

THE DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. GUPTA TECHNOLOGIES, INC. SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

This document may describe features and/or functionality not present in your software or your service agreement. Contact your account representative to learn more about what is available with this Gupta Technologies® product.

Gupta Technologies, Inc.  
1420 Rocky Ridge Drive, Suite 380  
Roseville, CA 95661

Gupta Technologies.com

# Table of Contents

## [TEAM DEVELOPER 6.2 SP3](#)

[RELEASE NOTES](#)

[SERVICE PACK PATCH INSTALLERS](#)

[SALLISTSETDROPDOWNDWIDTH](#)

## [TEAM DEVELOPER 6.2 SP2](#)

## [TEAM DEVELOPER 6.2 SP1](#)

## [TEAM DEVELOPER 6.2](#)

[IDE FEATURES](#)

[GUI FEATURES](#)

[Group By Order By Group by Summary](#)

[Hidden vs. visible columns Row Height](#)

[.NET NEW FEATURES](#)

[LANGUAGE API FEATURES](#)

## [TEAM DEVELOPER 6.1 SP3](#)

[BEHAVIOURAL DIFFERENCES](#)

## [TEAM DEVELOPER 6.1 SP2](#)

## [TEAM DEVELOPER 6.1 SP1](#)

## [TEAM DEVELOPER 6.1](#)

[IDE FEATURES](#)

[GUI FEATURES](#)

[GRID SUMMARY BAR](#)

[GRID ENHANCEMENTS](#)

[TABLE ENHANCEMENTS](#)

[TAB BAR ENHANCEMENTS](#)

[CQUICKGRAPH CHART CONTROL REPLACEMENT](#)

[SALRIBBONMAXIMIZE, SALRIBBONMINIMIZE](#)

[NAMED TOOLBARS](#)

[DATE PICKER ATTRIBUTE: WEEK NUMBER](#)

[TREE CONTROL ENHANCEMENTS](#)

[INTERNATIONALIZATION](#)

[.NET NEW FEATURES](#)

[UNUSED CODE/ORPHAN ANALYZER](#)

[WEB SERVICES](#)

[ENHANCED XAML SUPPORT](#)

[.NET EXPLORER ENHANCEMENTS](#)

[.NET MISCELLANEOUS ENHANCEMENTS](#)

[DATABASE CONNECTIVITY DBPIE](#)

[REPORT BUILDER INCLUDE SYSTEM FORMULAS](#)

## [TEAM DEVELOPER 6.0](#)

[BUILD .NET APPLICATIONS](#)

[ENHANCED GROUP BOX CONTROL](#)

[ENHANCED TAB BAR CONTROL](#)

[NAVIGATION BAR](#)

[OUTLINE EDITOR - INSERT TASK](#)

[QUICKTAB2TAB CONVERSION TOOL](#)

[REPORT BUILDER "NEW QRP" OPTION](#)  
[SALGRID EXPORT FUNCTIONS](#)  
[SALGRIDSETALTROWBACKCOLORS](#)  
[SALTBLCREATECOLUMNFROMCLASS](#)  
[SALMAIL - SMTP AND POP3 SUPPORT](#)  
[TREE CONTROL](#)

*[Note for SP3: Uninstalling TD 6.0 SP1 Full + SP3](#)*

#### **[TEAM DEVELOPER 5.2 SP1](#)**

[IMAGES IN POPUP MENUS](#)  
[PICTURE OBJECT IMPROVEMENTS](#)  
[MULTI-LINE TOOLTIPS AND COLUMN TITLES](#)  
[SALFILEOPENC](#)  
[UNICODE IN INFORMIX](#)

#### **[TEAM DEVELOPER 5.2](#)**

[CDK UPDATE](#)  
[COMPILER TAB IN PREFERENCES](#)  
[CONNECTION STRINGS](#)  
[DATE AND DATETIME MESSAGING](#)  
[EMBEDDED IMAGES](#)  
[GRID CONTROL](#)  
[HTML HELP](#)  
[MAX DATA LENGTH: CHARACTER & BYTE](#)  
[MULTIPLE OBJECT SELECT](#)  
[NEW BINARY DATA TYPE VARIABLE](#)  
[PUSH BUTTON AND OPTION BUTTON ATTRIBUTES](#)  
[REPORT BUILDER TEXT ROTATION](#)  
[RIBBON CONTROL ENHANCED](#)  
[RICH TEXT CONTROL](#)  
[SUPPORT IMPORTING WEB SERVICES WHICH REQUIRE AUTHENTICATION](#)  
[SUPPORT FOR LATEST WINDOWS VERSIONS](#)  
[TAB CONTROL](#)  
[TABLE2GRID UTILITY](#)  
[TOOLTIPS](#)  
[WARNING MESSAGE WHEN USING DEPRECATED API](#)  
[WATERMARK IMAGE SUPPORT \(REPORT BUILDER\)](#)

#### **[TEAM DEVELOPER 5.1 SP3 HOT FIX 3](#)**

[PERFORMANCE ENHANCEMENT](#)

#### **[TEAM DEVELOPER 5.1 SP3 HOT FIX 2](#)**

[ADDED TWO NEW SAL API'S](#)  
[NEW PDF SETTINGS DIALOG BOX IN REPORT BUILDER.](#)

#### **[TEAM DEVELOPER 5.1 SP1](#)**

[NEW FEATURES](#)  
[RELEASE NOTES](#)  
*[Some Patch details.](#)*  
*[Strict syntax type checking is now enforced.](#)*  
*[Using a theme other than the default themes provided.](#)*  
*[Reverting back to WinHelp](#)*

#### **[TEAM DEVELOPER 5.1 SP2](#)**

[NEW FEATURES](#)  
*[Better Win32 API Migration Support](#)*

[RELEASE NOTES](#)

[External DLL's with HSTRING's and UDV's.](#)  
[Sybase UNITEXT data type.](#)  
[ODBC longbuffer setting:](#)

**[TEAM DEVELOPER 5.1](#)**

[UNICODE SUPPORT](#)  
[WEB SERVICES SUPPORT](#)  
[CONNECTIVITY](#)  
[REPORT BUILDER PDF SUPPORT](#)  
[SUPPORT FOR THEMES](#)  
[DATE PICKER](#)  
[DATE TIME PICKER](#)  
[TAB CONTROL](#)  
[RIBBON BAR](#)  
[GRID CONTROL](#)  
[WEBHELP](#)  
[RELEASE NOTES](#)

**[TEAM DEVELOPER 4.2](#)**

[TEAM OBJECT MANAGER ENHANCEMENTS](#)  
[MISCELLANEOUS ENHANCEMENTS](#)

**[TEAM DEVELOPER 2005.1](#)**

[XML OPERATIONS](#)  
[UDV SERIALIZATION AND DESERIALIZATION](#)  
[XML SAMPLE APPLICATIONS](#)  
[FIND-AND-REPLACE ENHANCEMENTS](#)  
[REPORT BUILDER ENHANCEMENTS](#)  
[TABLE WINDOW ENHANCEMENTS](#)  
[ODBC ROUTER ENHANCEMENTS](#)  
[MISCELLANEOUS ENHANCEMENTS](#)  
[NEW RESERVED WORDS IN SQLWINDOWS](#)  
[LINUX COMPATIBILITY](#)  
[IMPROVED PERFORMANCE](#)  
[FIND-AND-REPLACE ENHANCEMENTS](#)  
[REPORT BUILDER ENHANCEMENTS](#)  
[ACTIVE CODING ASSISTANT ENHANCEMENTS](#)  
[OLE DB ENHANCEMENTS](#)  
[OTHER ENHANCEMENTS](#)  
[NEW RESERVED WORDS IN SQLWINDOWS](#)

**[TEAM DEVELOPER 3.1](#)**

[SQLWINDOWS CLIENT APPLICATION GUI](#)  
[SQLWINDOWS DEVELOPER IDE](#)  
[MORE SQLWINDOWS NEW FEATURES](#)  
[REPORT BUILDER ENHANCEMENTS](#)  
[ENHANCED ORACLE ROUTER](#)  
[WEB APPLICATION MANAGER \(WAM\) FOR LINUX](#)  
[SQLBASE VERSION 8.5 INTEGRATION](#)



# Team Developer 6.2 SP3

## Release Notes

Beginning with SP3, the fixes lists for service packs will be located in Team Developer's subfolder, *Release Notes*.

## Service Pack Patch Installers

Service Pack Patch Installers will no longer be used. Team Developer 6.2 SP3 will recognize whether or not a previous release of Team Developer 6.2 (including patch installs) is installed on the target machine.

A prompt to upgrade TD 6.2 will display "This setup will perform an upgrade of Team Developer 6.2. Do you want to continue?" If a release of Team Developer 6.2 is not installed on the target machine, then a full installation will occur.

## SalListSetDropdownWidth

bOk = SalListSetDropdownWidth (hWnd, nNum)

Sets the width of a Combo Box, Grid Drop Down List, or Table Drop Down List. The width cannot be set shorter than the initial and default size.

Parameters:

hWnd: Window Handle. The handle (or name) of object on which you want to set its drop down width.

nNum: Number. Size of the Drop Down List (in pixels)

Return Value:

bOk is TRUE if the function succeeds and FALSE if it fails.

# Team Developer 6.2 SP2

- Team Developer 6.2 SP2 now supports Sybase ASE 15.7.
- TD-20349: This is "as designed" that Combo box with Attribute Setting "Always Show List" = YES has border.
- TD-20582: The tab bar was completely re-implemented with TD 6.0. It now requires that the tab bar be sized large enough to contain the controls assigned to it. Controls outside the bounds of the container are not allowed.
- TD-20521: The width for SalListSetDropdownWidth is in pixels.
- TD-20333: SalStrLopASCII removes a single byte from the front of the buffer. This changes the boundary of the two byte UNICODE character so it appears as garbage when interpreted as UNICODE.
- TD-20402: MB\_SYSTEMMODAL does not block user interaction of all other windows. It just places a message box window in front.
- Same as MB\_APPLMODAL except that the message box has the WS\_EX\_TOPMOST style. Use system-modal message boxes to notify the user of serious, potentially damaging

errors that require immediate

- attention (for example, running out of memory). This flag has no effect on the user's ability to interact with windows other than those associated with hWnd.
- TD-18433: cListView Column Alignment property is updated. Left returns 0, right returns 1, center returns 2.
- TD-20796: There is a new API for setting Menu colors. See SalSetMenuColor.
- TD-20197: XMLLIB.APL now compiles in .Net. Note that the following functions are not implemented:
  - HTTPRequest class
  - XMLDocument.LoadFromSQL function
  - XMLDocumewnt.SchemaFromSQL function
  - XMLElement.LoadFromSQL function
  - XMLElement.SchemaFromSQL function
- TD-20652: Added "Transparent" attribute type for background color of push buttons and option buttons. Users need to type "Transparent" in that property.
- TD-20929: STRING\_Null, SalStrLop( ) now returns zero.

## Team Developer 6.2 SP1

- Named Toolbars : There are three new APIs added to dynamically delete, hide and show items on a named toolbar. See SQLWindows In-Build Help for more information on SalTBDeleteItem, SalTBShowItem, and SalTBHideItem.
- To see the fix to defect TD-19206, users need to re-create new instances for Grid classes.
- A new sal api, SalListSetDropdownWidth, is added to set the width of the combo box. Please check SQLWindows In-Build Help for more information on this API.
- A new cReport function, SetIntlNumberDisplay, has been created to enable/disable the 'Intl. Number Display' flag in a generated report.
- SalXMLSerializeUDV and SalXMLDeserializeUDV are now supported in .NET WPF projects.
- There is a new flag, TBL\_Flag\_AllowSortIncomplete, that enables Sorting functionality on a Grid using TBL\_filleNormal or SalTBISetRange.
- With the fix to defect TD-20191, named menus and items can be localized. Use StaticResource with the corresponding XAML to localize.
- With the fix to defect TD-19952, DotNet Explorer now supports .NET 4 interfaces.
- Team Developer 6.2 SP1 is certified for Windows 8.1.
- Team Developer 6.2 SP1 is certified using SQL Server 2012. TD applications are tested using ODBC, OLEDB, and Native Drivers with the SQL Server 2012 database.
- Team Developer 6.2 SP1 is certified using Oracle 12c. TD applications are tested using Oracle 11.2 client with Oracle 12c database.

## Team Developer 6.2

## IDE Features

### File open dialog

SQLWindows developer opens a new file by choosing File | Open from the menu, or by clicking on the open toolbar button. The Open File dialog comes up and all files with extension .ap? are displayed.

### Select multiple child objects

SQLWindows developers can select multiple GUI objects on form layout and can move the whole set of controls from one tab page to another and can quickly select the pertinent controls and re-assign them. It only allows assigning to a tab page if the control is already on a tab control.

### Ability to re-arrange new outline tabs

SQLWindows developers can drag the outline tabs left and right in the IDE so as to put them in whatever order is desired.

### Quick message navigation

Quick message navigation allows developers to quickly navigate to the appropriate SAM Message by double clicking the control in form layout. For example, developer double clicks on a button in the layout window, the IDE switches to the outline and highlights the SAM\_Click of the button and expands the node so that the user can quickly see and possibly edit the contents of the node.

### Filter messages in Attribute Inspector

There are two new check boxes provided in the Attribute Inspector which allows users to filter SAM message or Users messages.

### Auto-cleanup of compiler temp files

Auto-clean feature is added to the .NET compiler. This cleans any temp folders created under the user's Temp folder when closing SQLWindows IDE.

### “Find All” search feature for IDE

SQLWindows has a "Find All" menu item under Edit popup menu. This allows users to find all occurrences of a particular string in an application and to move easily between these locations. Selecting the Find All menu choice will open a modeless dialog which allows the user to search for a string. All found instances will be loaded into a grid. If you double-click on a row, the outline will display that location. The search can also be restricted by "case".

### Menu items for TD Community and Support Forums

There are two new menu items to the Help menu in SQLWindow's IDE right under the item Upgrade License. TD Community and Support Forums. Both menus open the appropriate web pages in a browser.

## GUI Features

### Progress Bar replacement for cMeter

SQLWindows provides a native progress bar control called Progress Bar which is a replacement for the old cMeter Visual Toolchest control. A Progress Bar control can be added

like any control. All new Progress Bar APIs begin with SalMeter. See SalMeter APIs in the in-build help.

## Grid Enhancements

- **Save and load grid state to file.** User SalGridWriteState() and SalGridApplyState() to save and read the Grid state to and from an XML file. The following information can only be saved and retrieved:

*Group By Order By Group  
by Summary*

*Hidden vs. visible columns Row Height*

- **SalGridDataExport().** The SalGridDataExport() function has been expanded so that if the path parameter is empty, then Team Developer runtime automatically pops up a "Save As" file chooser dialog box to allow the user to interactively select the output path at runtime.
- **Support background color for grid rows and cells.** There is a new API SalTblSetCellBackColor, that can set the color on a specific cell in the grid. The behavior matches VisTblSetCellColor().

## Tree Control Enhancements

- **SalTreeMoveItem().** The SalTreeMoveItem() function moves a tree node, and all its sub-nodes, to a new location in the tree control.
- **SalTreeFindItemByData().** The SalTreeFindItemByData() function allows user to search through the nodes of a tree control to find a node with a specific data value.
- **SalTreeSetItemImageFromBinary().** The SalTreeSetItemImageFromBinary() function allows the user to specify the image for a tree node by passing in image data as a Binary variable.

## Date Time Picker Enhancements

- Show non-month days property for Date Time Picker. There is a new property "**Show Non Months Days**", which allows users set either True or False.
- Show Today attribute for DateTimePicker. There is a new property "**Show Today**", which allows users set either True or False to control Today button.
- Show None attribute for DateTimePicker. There is a new property "**Show None**", which allows users set either True or False to control None button.
- **Ability to set date format for DateTimePicker control.** There is a new Format attribute for Date Time Picker control, similar to the one used for datafields that is set to the datetime data type. Existing SAL APIs , SalFmtGetPicture() and SalFmtSetPicture() , are enhanced to work with DateTimePickers.
- Show Week Numbers for DateTimePicker control. There is a new property "**Show Week Numbers**", which allows users set either True or False to enable or disable week numbers at runtime.

## Named Toolbars Enhancements

There are additional APIs for users to dynamically control and customize toolbar buttons. All new Named Toolbars APIs begin with SalTB\*. See SalTB functions for in-build help or use Named Toolbar.pdf for more information.

## MDI background gradient

There is a new attribute "Themed Background" for MDI Window to control themes. Whatever theme is selected by the user will affect the painting of the MDI's client area.

## Docking

SQLWindows now provides "Allow Child Docking" attribute to dialogs. This provides the ability to dock a dialog box inside another dialog box.

## Watermarks in new Chart control

Developers can set a watermark into a chart control through a new chart API function called `UfcChartSetWatermarkBitmap()`. See in-build help for more information on this API.

## Rich Ribbon Bar

Team Developer provides SQLWindows developer the ability to design rich ribbon menus natively. The overall focus of the new ribbon bar will be an emphasis on easy-to-use, quick-to-create 4GL ribbons. See Ribbon Menu under Books folder for full details on how to use this feature.

## .NET New Features

### XAML Spell Checking for Datafield, multiline-filed and rich text control

Developers can turn on the new Spell Check attribute for the field that needs spell check. This is .NET only feature. This feature is supported for English, German, French and Spanish regions only.

### 64- bit applications on .NET

When compiling in .NET build setting, SQLWindows now has the option of specifying the "bitness" of the outputted file (EXE or DLL). The choices are 32 bit, 64 bit, or "auto". See `DotNet.pdf` under books folder for more information.

### SAM\_NetException

Developer are now able to put a global `SAM_NetException` message in their app to catch exceptions which occur when calling methods in external .NET libraries while in win32 mode. By default, when an exception occurs during the invocation of an external .NET method, the `SAM_NetException` message will be sent to the Application Actions section. If there is code and the code either returns TRUE or has no Return statement, then that ends the error processing. If there is no `SAM_NetException`, or there is one and it returns FALSE, then the old processing continues.

### Debug Web Services and Class Libraries

SQLWindows now provides the mechanism to debug into external web services and class libraries. See `DotNet.pdf` under Books folder for more information on this.

### Load XAML resource dictionary dynamically

There is a new API `SalDictionaryLoad()` that allows users to dynamically decide on the appearance of their application and selectively load resource dictionaries based on user input or saved settings.

### Assembly sign to register in GAC

SQLWindow provides the ability to enter the path to a Strong Name Key file (\*.snk) on the new Advanced Settings dialog box under Build Settings dialog. If this value is provided, the .NET compiler will attempt to sign the assembly using the file. Strong signing is needed to register an assembly in the GAC

### Custom exceptions

SQLWindows offers developers the ability to define their own custom exceptions and to throw them ad-hoc. Using the new language keyword "Throw", developers are now able to implement their own structured exception handling logic. This feature can also be used if the developer has a DLL that wishes to share with other applications and want to define a specific set of exported exceptions to govern error conditions.

Under Global Declarations, there is a new node called "Named Exceptions". The legal child-node of Named Exceptions is an "Exception" node. The Exception node itself just defines a simple string name, for example Exception: MyException.

Indented under the exception node, there is a standard "Description" node and an "Exception Variables" node. These nodes describe the data which can be specified at throw time to further describe the exception which occurred.

There is a new language keyword "LastException" to return the name of the last exception which occurred and also a new function SalGetLastExceptionField() to retrieve data from within the exception. The system variable "LastException" is not reset until another exception occurs. It is not a valid coding practice to check if an exception has occurred by constantly comparing this variable to an empty string.

## Language API Features

### Allow dynamic setting of web service address

There is a new SAL function SalWSSetEndpointAddress() to override the web service address which is embedded in the WS interop assembly which gets generated by the Team Developer Web Service Wizard.

### Convert strings into binaries

There is a new Sal API function SalStringToBinary() that takes a string value and converts it into a binary value in the new Binary datatype. This is a front end function to SalStrToMultiByte(). See in-build help for more information.

### Convert binaries into strings

There is a new Sal API function SalBinaryToString() converts a binary value into a string value. This is basically a front-end function to SalStrToWideChar(). See in-build help for more information.

### SalGetEnvironment

There is a new SalGetEnvironment() function that returns system information about the current machine, like OS version and hardware specifications, etc. Due to differences in .NET and Win32, not all the same keys are guaranteed to come back from SalGetEnvironment() when calling in Win32 and .NET. See in-build help for more information.

### Convert local times to and from Universal Coordinated Time (UTC)

There are two new Sal functions: SalDateToUTCtime() and SalDateToLocalTime(). These will take a date/time variable and convert it to UTC and vice-versa. The return value is a Date/Time variable. See in-build help for more information.

### Compiler warnings for GUI-related SAL functions within a DLL or Web Service

SQLWindows compiler gives warnings when GUI-related SAL functions are called within a DLL or Web Service build settings.

### SalDlgOpenFileMulti()

This new API works basically the same way as SalDlgOpenFile() except that users will be able to select more than one file. The two parameters describing the file which was picked will become arrays and a new argument is added at the end, giving the number of files which

were chosen. See in-build help for more information on this.

## Report Builder

### New Graph Control

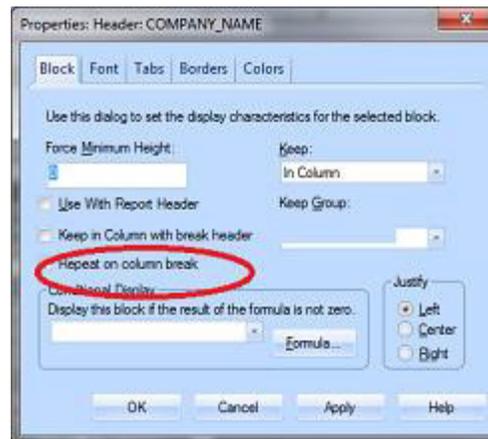
The existing support for embedding QuickGraph controls in a Report Builder report is replaced with the new Chart control. All existing QuickGraph properties are migrated forward and new Chart properties will be persisted within the QRP/CQT. Any existing QRP/CQT files will open and run in Report Builder 6.2 without modification, with the new Chart showing up instead of legacy QuickGraph. The new Chart properties dialog allows the user to set the desired report properties.

### Repeat Column Header after Break

When group contents are longer than the height of a column, a column break will appear. The user has an option to display the Column Header for that groups contents and any subsequent group contents after all column breaks in a report. In the Report Design, Select the column 'Header'. In the Tools toolbar, select the Report tab. Select Block (under Format items). Select (check) on 'Repeat on Column break'.

Or

Right-click on column 'Header' (context menu), select Properties....



## Team Developer 6.1 SP3

With the fix to defect TD-19166, the connectivity performance issue introduced in the previous SP3 release is now resolved. Note that the performance issue is caused by the SQLBase API (sqlwntm.dll) that is included in the Team Developer installation directory. The resolution for this is to revert back to the previous SQLBase API.

With the fix to defect TD-16120, the group and sort menu on the grid appears only when all rows are fetched. TBL\_FillAll populates an entire table window at once thus group and sort are enabled. TBL\_FillNormal populates the rest of the table window as you scroll and bring new rows into view. The group and sort menu on the grid is disabled.

## Chart Control

Team Developer Chart control now supports new Sal APIs for more customization. Check inbuilt help for `UfcChartSetAxisYLabelWidth`, `UfcChartSetLegendSize` and `UfcChartSetCustomLineLabel`.

## DBPipe

Team Developer now provides new APIs to add and retrieve headers from DBPipe to make it work in a WebSeal environment. Check in-build help for `SqlDbPipeSetRequestHeader` and `SqlDbPipeFetchResponseHeader`.

## Behavioural Differences

With the fix to the defect TD-17327, Grid Dropdown list column now supports inplace Combo Box editor.

With the fix to the defect TD-18621, for multi-line columns, the height of the Grid row with non-default font size is reduced by four pixels for each line. Note that the default font size for grid window is MS Sans Serif.

With the fix to the defect TD-18677, Team Developer no longer supports the call for windows APIs `BeginPaint()` and `EndPaint()` to get the Device Context. The workaround for this is to call `GetDC()` and `ReleaseDC()` functions. `GetDC()` returns the exact same Device Context as that of `BeginPaint()` function. Refer to `ClockWindow.app` shipped with latest Team Developer samples on how to change. Caution, if the application has defined the return value for `BeginPaint()` as a Window Handle, then `GetDC()` needs to be returned as a Window Handle. The problem with `BeginPaint` is it synchronously causes a `WM_ERASEBKGND` to be put in the message queue which does not get handled until the interpreter leaves the `WM_PAINT` handler in the Sal code. This causes painting to be erased. Internally, we handle these differently.

TD-16347 : Tooltip for Ribbon menu items displays the accelerator if defined. In order to show a more detailed tooltip for menu items, use Status Text field in menu editor. The default menu text gets displayed as menu item tooltip when there is no accelerator or status is defined.

In order to view the fix to defect TD-18895, users must re-deploy the application and re-install the Report Builder Embedded deployment (RBDeploy) on the server machine.

TD-19101 : A picture's background painting (frame and background color) no longer have a higher precedence than a frame background painting. Change the order in the outline if needed. This change in behavior is because of the fix to defect TD-17390.

TD-18807 : When using `SalReportPrintToFile()`, it is recommended to use PDF instead of Rich Text Field for standardized printing. The reason is RTF has a loose standard. The behavior often depends on the RTF reader. RTF does not provide precise positioning and features like line style and width.

TD-19007 : The fix to this defect causes the removal of two extra bytes when selecting binary from a database.

## Team Developer 6.1 SP2

TD-17467: A Wrong character displays in the Outline Description and when using multiple line breaks with ctrl-Enter. This defect was fixed in TD6.1 SP1 but introduced a regression so it had to be back ported. Upon further investigation, the Gupta Development team found that this defect was caused by the virtual machine software and cannot be fixed by Daegis. Therefore, users of virtual machines will see the wrong character in the outline description and when using multiple line breaks with ctrl-Enter. That wrong character will not interfere with the developer's ability to edit the application nor corrupt the outline.

## Team Developer 6.1 SP1

The fix for defect TD-15122 causes a change in behavior for the following functions when used with Binary variables:

SalGetBufferLength

SalPicGetString

SalPicGetImage

SalFileRead

SalActiveXGetData

The above functions when used with Binary variables will no longer contain a NULL Terminator (2 Bytes).

Sql statements using binds and into as binary variables do not add or remove terminating nulls. Note that this change has no affect on Long String variables.

For Data Source connectivity using ODBC DSN, gupta.ini is generated into the users TEMP folder.

The fix for defect TD-15052 makes the compiler strict and returns an error if there is a mismatch in parameters.

For Example:

Sample VTTTest.app (Team Developer 6.1 RTM version) will return a Function argument error when compiled with Team Developer 6.1SP1.

SalListClear must specifically call the Window Handle of the control(list box) to correct this error.

With the fix to defect TD-17318, Option Buttons now honors the "Button appearance" property.

# Team Developer 6.1

## IDE Features

### Color-coded Source Code

The source code in the IDE is now color-coded. You can customize the colors of each of the following elements:

- String Literals
- Include statements
- Comments
- Keywords

Configure the color-coding in the Tools | Preferences dialog under the Outline tab.

### Background Text & Group Box "Object Title" displayed in outline

To help identify background text and groupbox controls in the outline, the "Object Title" attribute is now displayed after the control name. This is for display only and is not editable via the outline. You can change the Object Title using the Attribute Inspector.

Additionally, it is now possible to assign a string to a background text or groupbox. Example: Set bkgd1 = "Last Name"

### nVar++, nVar--

The active coding assistant has been enhanced to accept this syntax for incrementing or decrementing numeric variables. The code will automatically change to acceptable SAL code, such as "Set nVar = nVar + 1."

### Go To Declaration

A "Go To Definition" item has been added to the right-click menu in the IDE outline. Right click on a function, variable, or class instance and select this item to see where the item you clicked is defined.

### Multistep undo

Now the Edit | Undo command can undo 5 actions.

### Tab Bar (in IDE) for Multiple Outline Views

A tab bar has been added to the top of the IDE window to allow for multiple outline views to be open simultaneously. Use this feature to navigate between sections of code without searching for their context in the left panel.

### Double-click Navigation for default messages

Double-clicking on a control in the layout window automatically takes you to the "default" message for that control type in the outline view. For example, double-clicking on a PushButton takes you to the SAM\_Click message for that button.

## Easier message navigation

The Attribute Inspector now has tabs at the bottom. Click on the messages tab to display all the valid messages for the currently selected control. If a message action has been coded for the control, the message is shown in bold. Double-clicking on a message takes you to that message in the outline.

## Active Coding Assistant

The Active Coding Assistant has been completely redesigned to maximize performance in large applications. It is also resizable, giving the user more control over the listed content.

The Active Coding Assistant features an AutoComplete control providing front-end logic for text entry suggestions and auto-completion. This includes inherited class members and imported symbols from AXL files.

## Documentation Creator

This tool allows you to generate detailed HTML documentation from the description comments in your code. To launch this tool, go to **Tools | Create Documentaton**. It is highly customizable and allows you to easily apply your own stylesheet designs.

The homepage file is specified by the "Homepage File" setting on the main screen. All other files are placed in a subfolder specified by the "Subfolder Name" setting. One html file is generated for each class, form, and function in the application. The files of the application which will be included in the documentation can be selected on the main screen with the "Include Files" listbox. In this way, you can filter out third party code from the documentation. Detailed "layouts" can be created and saved under seperate names using the Layouts listbox. Each defined layout allows you to specify the text of the section headings, header and footer html for each page, and the name of any \*.css files to use.

## Debugging Enhancements

- Hover over UDV's and UDV arrays to view current values.
- Message debugging window now displays detailed information about each message, including Message, Class, Window, wParam, lParam, and Handle.
- A stack window has been added for .NET apps.

## GUI Features

### Grid Summary Bar

The Grid has been enhanced to support a Summary Bar. This feature, when enabled, inserts a summary row at the bottom of the grid. This self-maintaining row can contain the following statistics for each column: total, maximum, minimum, and average. Statistics can be set per column using the Attribute Inspector.

The summary bar can be enabled or disabled by using the new Grid attribute **Summary Bar Enabled**. Programmatically, you can use **SalGridSummaryBar** to enable/disable the summary bar.

The statistic to display for each column can be set per column using the new attribute **Column Aggregate Type** in the Attribute Inspector. Programmatically, the statistics can be changed using **SalGridSetSummaryColumn**:

```
Bool SalGridSetSummaryColumn(Grid,Column Index,Statistic Type)
```

- Grid: Window handle to grid.
- Column Index: The number of the column to modify.
- Statistic Type: 1 = Minimum, 2 = Maximum, 3 = Sum, 4 = Average

You can also place a label on your summary column with **SalGridSetSummaryColumnLabel**(Grid, Column Index, Label)

## Grid Enhancements

- **SalGridDataImport** - enables the user to programmatically populate a grid from Excel, CSV, TXT, or XML files.
- **SalGridSetRowHeight** - enables the user to set the height of a grid's row based on a percentage of the normal height.

## Table Enhancements

- **SalTblDefineRadioButtonColumn** - enables the user to define a column as a radio button column and set its TRUE and FALSE values.

## Tab Bar Enhancements

- The new Tab Bar Control supports multiple rows of tabs.
- **SalTabAddPageEx** - Allows the user to add an icon and tooltip to a tab page created at runtime.
- **SalTabGetName** - Retrieves tab name based on provided tab index.

## cQuickGraph chart control replacement

The QuickGraph control is now based on a modern WPF control with greatly expanded features and functionality. All existing instances of the cQuickGraph control are automatically migrated to the new chart control, and all properties of the new chart are accessible at design time via a new properties dialog.

## SalRibbonMaximize, SalRibbonMinimize

These functions allow the user to programmatically maximize and minimize the ribbon bar. See **SalRibbonMaximize** and **SalRibbonMinimize** in the in-build help.

## Named Toolbars

Applications can include multiple dockable toolbars like those in the Team Developer IDE:



In the outline, under the Global Declarations section, you will find **Named Toolbars**. Like Named Menus, Named Toolbars are defined once and can be created in multiple locations. These floating, dockable toolbars facilitate more modern, customizable applications.

Each Named Toolbar may contain a mixture of buttons and dividers. Each button has an **Actions** section, which contains the code to be run when the button is clicked. A button may display an image, specified by the **Picture File Name** property. If no image is specified, the button's name is displayed.

Named Toolbars are created using **SalCreateToolbar**:

Bool SalCreateToolbar(owner,toolbar name, position,x,y)

- Owner: Window handle of owning form.
- Toolbar name: The named toolbar to create.
- Position: The docking location for the toolbar:

DOCK\_Top: Dock to the top of the form DOCK\_Bottom: Dock to the bottom of the form.

DOCK\_Left: Dock to the left of the form.

DOCK\_Right: Dock to the right of the form. DOCK\_None: Free float the toolbar

## Date Picker Attribute: Week Number

A new attribute, **Show Week Number**, has been added to the Date Picker. When this attribute is set to "Yes," the Date Picker displays a number (week counter) in the left margin of each week.

A new SAL API, **SalDateWeekNumber**, has also been added. Given a date/time value, it returns the number of the week for that date.

## Tree Control Enhancements

- Support for **SAM\_ContextMenu**. The user may trap and respond to right clicks on a Tree Control.
- **SalTreeReset** - Clears a tree control with the option to retain or destroy design-time nodes.
- **SalTreeItemFromPoint** - Given x/y coordinates, returns hItem of tree item at that location.

## Internationalization

A new attribute has been added for field controls called **Flow Direction** which allows the field controls to support left-to-right or right-to-left languages.

## .NET New Features

### Unused code/Orphan analyzer

This new feature, accessed via **Project | Analyze**, is available for .NET build targets. It performs a recursive analysis of your code to identify forms, classes, variables, and functions that are not used anywhere in the application. Code must be free of compilation errors before running the analyzer tool.

The determination to mark an item as "used" is scope-based. In other words, a code item may be used somewhere in the application, but if it is only used outside of its own context, then the item itself and its reference are considered unused. In addition, deleting an unused item could result in a compile error. For example, function A may be listed as unused, but deleting it may result in a compile error because function A is called by function B. However, function B is not used anywhere in the code, so deleting function B will allow the application to compile again. Thus, care should be taken when removing an item marked as unused.

## Web Services

When in .NET mode, a new web service wizard allows you to import a WSDL file, create a .NET interop assembly for the web service, and import the symbols (\*.AXL) into your application for easy .NET web service consumption.

Additionally, Team Developer 6.1 supports the creation of .NET Web Services. A new class type, Web Service Class, allows you to export your non-GUI logic as a web service. The new function type operation allows you to distinguish between those methods which are exported and those which remain internal to the web service.

Finally, **SalThrowSoapFault** has been added for throwing a soap fault.

## Enhanced XAML Support

Team Developer 6.1 includes a new global resource: **Brushes**. These are designable using the new Brush Editor and assignable to any background or foreground color of any control.

Another new global resource in this version of Team Developer is the **Resource Dictionary**. You can include a custom Resource Dictionary directly into your application without editing the app.xaml file.

Controls have also been enhanced to include a **XAML Style** attribute which can be used to assign a custom style to each control.

Finally, **SalDictionaryPromote** has been added for customizing the precedence of resource dictionaries.

## .NET Explorer Enhancements

- Support for multiple generated APLs.
- Warning when generating a previously generated APL.
- User-friendly .NET Explorer window.

- You can directly select whether to import an APL or an AXL, depending on your build target. For AXL imports, duplicate symbols between AXL files will be ignored silently by the .NET compiler.

## .NET Miscellaneous Enhancements

- Team Developer 6.1 offers the ability to compile your application in .NET 4.
- Local .NET Exception Handling - The new language statement `When Exception` allows you to trap exceptions locally within a function. The design is similar to `When SqlError` in that you first specify the `When Exception` node, and then all following code that lies at the same or greater indentation will be covered by this exception handling. Unlike `When SqlError`, there is no concept of using a `Return` statement to modify the returned value of the function that caused the exception. Also, execution does not continue at the point just after the exception occurs, but instead continues at the next less-indented level from the `When Exception` statement. The new statement `Rethrow`, allows you to "throw" the exception up to the next level. It will then be caught at the most recent `When Exception` contained on the call stack. A special version of `Rethrow`, `Rethrow Global`, allows the exception to be thrown straight to the global exception handler (i.e. `SAM_NetException`). In this way, the user can mimic a two-tier error handling system as used historically for SQL trapping.
- Application namespace can now be set in the build dialog.
- Structures are now supported as parameters to external functions.

## Database Connectivity DBPipe

DBPipe usage has been enabled for WPF desktop applications. In addition, the server component has been given a UI with enhanced performance tuning and logging abilities.

## Report Builder Include System Formulas

ReportBuilder automatically generates a formula for each column of data in a query. You can now specify whether you want to include these system formulas in the tree view (left panel in design mode). They are hidden by default. To display them, go to Report Preferences and check the box labeled "Include System Formulas."

# Team Developer 6.0

## Build .NET Applications

Team Developer applications can now be built as .NET applications simply by changing the build settings (Project Menu / Build Settings). Continue programming in TD and roll out your applications in the .NET platform!

*See Introduction to .NET Projects on page 1-1 of the document entitled .NET Projects.*

## Enhanced Group Box Control

The Group Box Control has three new attributes:

- Line Color - Customizes the color of the groupbox border.
- Line Thickness - Sets the thickness of the border (1-8).
- Group Box Style - Provides five style options

See *Group box* on page 5-36 of the document entitled *Developing with SQLWindows*.

## Enhanced Tab Bar Control

The tab bar control has been enhanced in several ways. Here are a few examples of design-time enhancements:

- Click through tab pages in the layout view - child objects are displayed on each page.
- Associate and disassociate child objects by dragging and dropping controls in the layout view.
- Utilize 8 new SalTab functions.

See *Tab Bar* on page 5-60 of the document entitled *Developing with SQLWindows*.

## Navigation Bar

A navigation bar control has been added to SQLWindows. A common example of this type of control is the left-hand side of Microsoft Outlook. SalNav functions have also been added for runtime manipulation of the navigation bar.

See *Navigation Bar* on page 5-41 of the document entitled *Developing with SQLWindows*.

## Outline Editor - Insert Task

Right click on an item in the application outline and select "Insert Task." This feature allows you to add commented tasks that automatically include a date/time stamp.

See *Commented Items and Tasks* on page 2-10 of the document entitled *Developing with SQLWindows*.

## QuickTab2Tab Conversion Tool

Convert your QuickTabs to enhanced native Tab Bar controls using this conversion tool. Converting to native Tab Bars allows you to roll out your application in the

.NET platform.

See *QuickTab2Tab Tool* on page 2-2 of the document entitled *.NET Projects*.

## Report Builder "New QRP" Option

Report Builder typically assumes that you are creating or editing a .CQT file, which must contain a query (not just a report template). The "New QRP" File Menu item enables developers to quickly create a report template (.QRP file) that does NOT require a query.

See *Setting up the report template* on page 4-3 of the document entitled *Report Builder - Business Reporting*.

## SalGrid Export Functions

You can now export the contents of grid windows to .xls, .txt, .csv, and .xml files using five new SalGrid functions:

- SalGridDataExport
- SalGridSet/GetFooter
- SalGridSet/GetHeader

See the in-build help for more information on these functions.

## SalGridSetAltRowBackColors

This function enhances the grid control in WPF applications. Use it to set alternating background colors for the rows in the grid.

For details, see SalGridSetAltRowBackColors in the in-build help.

## SalTblCreateColumnFromClass

This function has been added to enable developers to create a new column using a column class as a template. The column can be inserted in any position of any Table Window or Grid.

For details, see SalTblCreateColumnFromClass in the in-build help.

## SalMail - SMTP and POP3 support

The SalMail class has been added to provide applications with email capability via SMTP and POP3 protocols. SalMail uses POP3 to receive emails, and SMTP to send. The cQuickEMail class is still available for accessing mail systems that conform to MAPI (Microsoft Mail).

For more information and a list of SalMail functions, see "SalMail Class" in the in-build help.

## Tree Control

A tree control has been added for the representation and navigation of hierarchical data. The control can be customized at design time via numerous object attributes, and runtime manipulations can be accomplished through dozens of new SalTreeXXX functions.

For more information, see *Tree Control (and tree items)* on page 5-68 of the document entitled *Developing with SQLWindows*.

### Note for SP3: Uninstalling TD 6.0 SP1 Full + SP3

There is a known issue when uninstalling this combination of service packs. To avoid the issue, run the batch script **manuallyuninstall.bat**, which is located at the root level of your Team Developer installation. This will successfully uninstall the software.

# Team Developer 5.2 SP1

## Images in Popup Menus

A "Picture File Name" attribute has been added to Popup Menus. When you select a popup menu in the hierarchal 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 2-20 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.

# Team Developer 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:

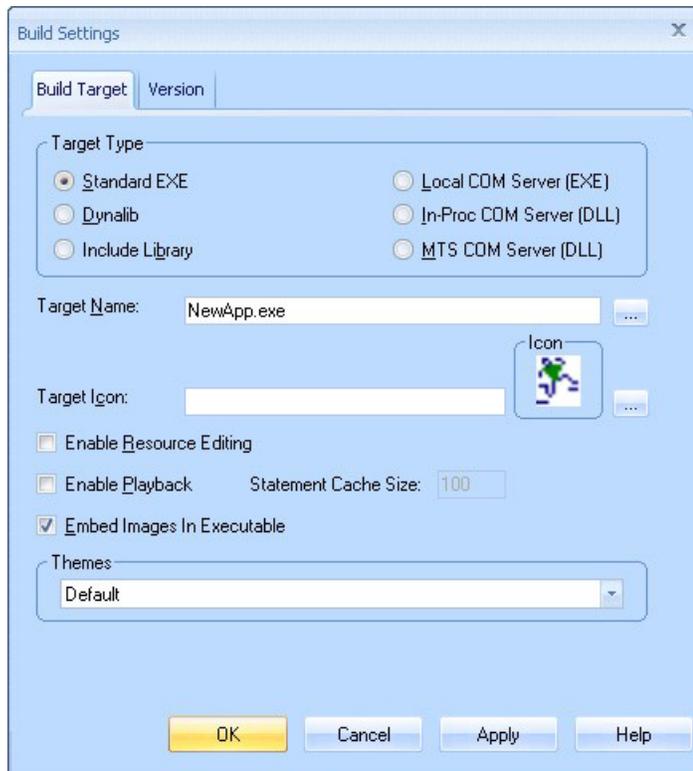
SAM\_AnyEdit - To capture any keystroke

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 apis. For information on the Grid Control, including sorting and grouping, see *Option button* on page 5-43 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 2-22 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 5-50 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 5-55 and *Controls palette* on page 2-29 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 apis.

## Warning message when using deprecated api

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*.

# Team Developer 5.1 SP3 Hot Fix 3

## Performance Enhancement

This is 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

# Team Developer 5.1 SP3 Hot Fix 2

## Added two new Sal API's

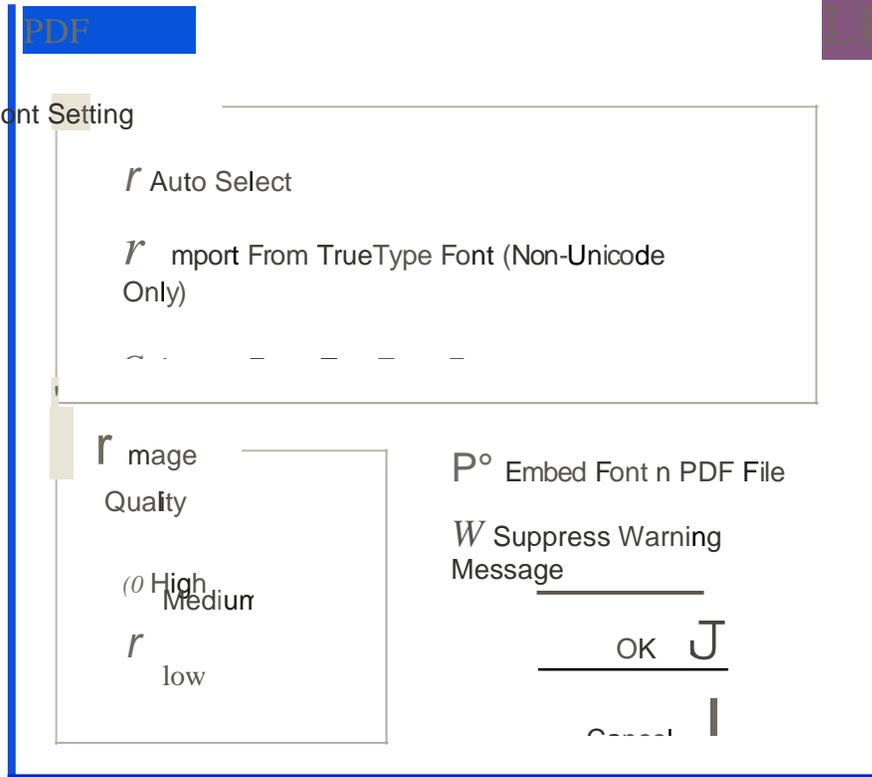
The following new sal api's were added:

- 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.



# Team Developer 5.1 SP1

## New Features

Team Developer 5.1 SP1 was strictly a service pack; no new features were added with this release.

## Release Notes

Following are the release notes for Team Developer 5.1 SP1.

### Some Patch details.

- Once Team Developer 5.1 SP1 patch is installed you cannot rollback to Team Developer 5.1
- The SP1 MSP patch has been archived into an executable.
- Web Application Manager was not installed in base release of Team Developer 5.1 and is now automatically installed with the patch. Also it is registered as a windows service.

### Strict syntax type checking is now enforced.

Team Developer 5.1 and above is now unforgiving if you do not declare parameters or return types correctly.

### Using a theme other than the default themes provided.

In order to choose a theme other than the default themes provided you will need to do the following:

- Install the new theme supported by the OS. In XP you need to copy the new theme to the directory: ..\WINDOWS\Resources\Themes\
- Open the "Display Properties" dialog. (i.e. go to the desktop, right click and select properties)
- Select the theme you just added in the "Theme" combo box and click OK.
- In the Team Developer IDE click on the "Native Windows XP" theme in the build settings dialog and build the application.

### Reverting back to WinHelp

Due to the negative response to WebHelp it was decided we would revert back to WinHelp. WinHelp is much faster and provides context sensitive help. The initial thinking behind going to WebHelp was because WinHelp was not being supported in Vista. For Vista users, you will need to download WinHlp32.exe from the Microsoft download site.

# Team Developer 5.1 SP2

## New Features

The following was added as a new feature in Team Developer 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*.

## Release Notes

Following are the release notes for Team Developer 5.1 SP2.

### External DLL's with HSTRING's and UDV's.

A new section of documentation was added to the "Developing with SQLWindows" document that talks about this topic. See External DLL's with HSTRING's and UDV's

### Sybase UNITEXT data type.

The Sybase data type UNITEXT is not supported in this release of Team Developer.

### ODBC longbuffer setting:

Make sure that the size of the ODBC longbuffer setting in the sql.ini file is less than or equal to the largest allowable size of the TEXT or image data type of the respective database. For SQL Server, this is 2GB's for a TEXT/IMAGE data type. If this value is greater than this allowable size, the database will run into the following error:

"Error 20029: "[Microsoft][SQL Native Client] String data, length mismatch"

# Team Developer 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 has 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.

## 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, Option button*

## WebHelp

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

## Release Notes

Following are the release notes for Team Developer 5.1:

### Changes in structure mapping external functions

There is a situation where a string is mapped to a byte[] in an external function. We do provide char[] mapping in an external function and this can be treated as ASCII data. In Team Developer 5.1, since a string is unicode, we need to convert the actual data to a unicode string. If the external structure is mapped to char[], Team Developer runtime will assume that the data is in ascii or local code page and do the appropriate conversion. If it is mapped to byte[], then the data will be considered as binary and no conversion will be done.

### Include library and Breakpoint details:

Old serialized data for include library and breakpoint details are discarded while migrating pre-5.1 applications. Pre 5.1 binary applications will loose breakpoint details after migration.

## Binary data binding:

Some users may have stored binary and text data in a BLOB datatype in SQLBase. Since there is no way to identify the content of a BLOB column in SQLBase, Team Developer runtime can not convert the text data to unicode data when users bind string variables to BLOB column.

In Team Developer 5.1, we treat the BLOB column as a true binary type and retrieve the data as is (i.e. no conversion at runtime). This could break the old application and this can be avoided by following these methods.

## Migration:

Method 1: User is to convert the data to unicode using the `SalStrToWideChar` api after data is retrieved from the database.

Method 2: Users should change the column type from BLOB to some type which supports unicode. This will trigger Team Developer runtime to do the appropriate conversion.

When migrating applications to Team Developer 5.1 it should be standard operating procedure to first perform a backup of your application and associated files before migrating. Also ensure that the files involved have write permissions before migrating.

## Opening a 5.1 application in a previous version:

If you want to open a Team Developer 5.1 application in a previous version of Team Developer the following steps need to be followed:

- Open your Team Developer 5.1 application and save as an apt.
- Open this apt in notepad.
- Change the outline version (i.e. 4.0.37 ) to the corresponding outline version for the Team Developer version you want.

The table below lists the outline versions which corresponds with various Team Developer versions.

Team Developer Version	Outline Version
1.1.x	4.0.26

Team Developer Version	Outline Version
1.5.x	4.0.27
2.0.x	4.0.28
2.1.x	4.0.28
3.0.x	4.0.31
3.1.x	4.0.32
4.0.x	4.0.34
4.1.x	4.0.35
4.2.x	4.0.35

- Choose 'Save as...' option in Notepad.
- Select "ANSI" in the encoding combobox.

- Save the file.

## SalStrBufferLength

nLength = SalStrGetBufferLength(strString) nLength is the length of the strString buffer in bytes.

Team Developer uses 16bit characters from version 5.1 onwards, all the Team Developer applications developed with prior versions of Team Developer using this API will show a change (i.e. increase) in the function's return value.

## SalTbIPopulate

SalTbIPopulate does not support the display of long column data for OLE DB connectivity.

---

**Note:** This is a limitation from previous versions of Team Developer.

---

## SqlSetLongBindDatatype

It is now mandatory to call "SqlSetLongBindDatatype(bindvarindex,bindtype)" in applications where there is code that binds blob data using long string variables. This needs to be done on both insert and fetch operations.

## Inserting Unicode data in SQL Server

When inserting unicode data to a Microsoft SQL Server database you need to prefix an "N" to denote that you are passing unicode data. Otherwise that data would be treated as ASCII by your database. Example: insert into tableX(fname,lname) values (N"Peter",N"Ryan")

## Passing Unicode data with Oracle

With the Oracle database you need to set the NLS\_LANG environment variable on the client side in order to pass unicode data.

---

**Note:** With Team Developer 5.1 SP2 and above this is no longer a requirement.

---

## Binding

It is important to use the right data type when binding columns containing string data. For example you need to use a string variable when binding to a char/varchar/nchar/ nvarchar column and use long string variable when binding to long column types such as long/long varchar/long nvarchar etc.

## Compiler Support for Extended Symbols

If you want the compiler to support the following extended symbols:

- 2460 ... 24FF (i.e. Enclosed Alphanumerics)
- 3000 ... 303F (i.e. CJK Symbols and Punctuation )
- 3200 ... 32FF (i.e. Enclosed CJK letters and months)
- FF00 ... FFEF (i.e. Halfwidth and Fullwidth Forms)

You will need to add a new DWORD value called **ExtCompilerChars** with a non zero value to the windows registry key HKEY\_CURRENT\_USER\Software\Gupta\SQLWindows 5.1 \Unicode

---

**Note:** You are encouraged not to use these types of characters in your applications as it will slow down compile time. When we talk about use we mean anything which the compiler will compile such as variables, constants, function names etc.

---

# Team Developer 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 HttpRequst\_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.

---

# Team Developer 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 SaXMLSerializeUDV, SaXMLDeserializeUDV, and SaXMLGetLastError. 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 and 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_INFOMIXONLINE` 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

## 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>	

# Team Developer 3.1

## 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.

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

New messages include:

Message	Description
SAM_WriteXMLRow	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

## 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

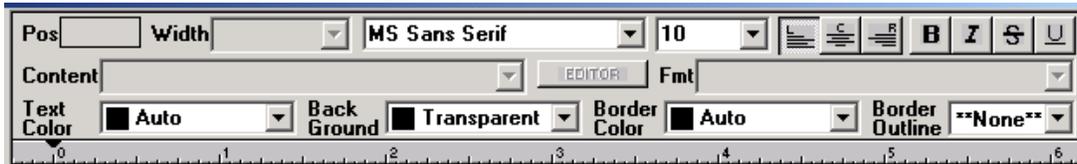
report and a different tray for all other pages. These preferences automatically adjust for the printer selected. If you change printers and the selected tray is not available on the new

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

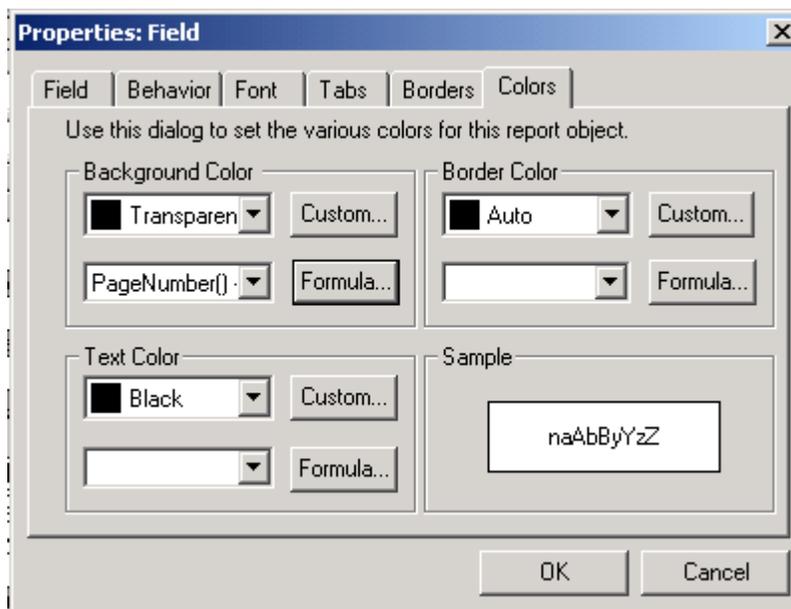
New controls at the bottom edge of the ribbon toolbar allow you to choose text color, background color, border color and border outline for report objects without invoking their property dialogs.



## 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 **SqIINI**, 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 SqIINI variable, Team Developer will use the method of locating SQL.INI that was used in previous versions.

Variable SqIINI, once changed, affects all future connections, through both SqlConnection and SqlCreateSession. If a connection is already open when SqIINI 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 SqIINI.

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