

TEAM DEVELOPER™

New & Changed Features

Product Version 7.0

Team Developer™: New & Changed Features

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111 Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440

Fax: +1-519-888-0677

Support: <http://support.opentext.com>

For more information, visit <https://www.opentext.com>

Copyright © 2016 Open Text. All rights reserved. OpenText is a trademark or registered trademark of Open Text. The list of trademarks is not exhaustive of other trademarks, registered trademarks, product names, company names, brands and service names mentioned herein are property of Open Text or other respective owners.

Disclaimer

No Warranties and Limitation of Liability. Every effort has been made to ensure the accuracy of the features and techniques presented in this publication. However, Open Text Corporation and its affiliates accept no responsibility and offer no warranty whether expressed or implied, for the accuracy of this publication.

Warning: This software is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Table of Contents

TEAM DEVELOPER 7.0	6
64 BIT SUPPORT	6
REST API.....	6
GUI CONTROLS	6
LANGUAGE.....	7
DATABASE	8
CDK	8
.NET EXPLORER.....	8
OUTLINE.....	8
TEAM DEVELOPER 6.3 SP1 UPDATE 2	8
TEAM DEVELOPER 6.3 SP1 UPDATE 1	8
TEAM DEVELOPER 6.3 SP1	9
TEAM DEVELOPER 6.3 UPDATE 2	9
TEAM DEVELOPER 6.3 UPDATE 1	10
TEAM DEVELOPER 6.3	10
LOOK AND FEEL.....	10
<i>PNG and JPG Files.....</i>	<i>10</i>
WEB REPORTING	10
<i>SalWebReportCreateToFile.....</i>	<i>10</i>
<i>SalWebReportCreate.....</i>	<i>11</i>
RIBBON BAR.....	12
<i>Ribbon Gallery Control.....</i>	<i>12</i>
<i>SalRibbonAddItem.....</i>	<i>12</i>
<i>SalRibbonDeleteItem.....</i>	<i>13</i>
<i>SAM_RibbonShow.....</i>	<i>13</i>
<i>SalRibbonPopulate</i>	<i>13</i>
<i>SalRibbonSetItemImage</i>	<i>14</i>
<i>SalRibbonSetItemPicture.....</i>	<i>14</i>
<i>SaRibbonSetItemSize</i>	<i>14</i>
<i>SalRibbonSetItemVisible.....</i>	<i>15</i>
GRID.....	15
<i>Grid Column Summaries for Groups.....</i>	<i>15</i>
<i>SalGridGetSummaryColumnValue & SalGridGetSummaryColumn.....</i>	<i>15</i>
<i>SalGridSetSummaryColumnJustify</i>	<i>16</i>
<i>CellProp_ReadOnly.....</i>	<i>16</i>
<i>Popup Edit.....</i>	<i>16</i>
<i>CellProp_Image_HeaderFromRes & CellProp_Image_FullFromRes.....</i>	<i>16</i>
<i>SalGridDataImport</i>	<i>16</i>
SALMAIL	17
<i>SendMailEx.....</i>	<i>17</i>
DATE PICKER/DATE TIME PICKER	17
<i>SalCalGetSelectedDates</i>	<i>17</i>
<i>SalCalMarkDays.....</i>	<i>17</i>
RICH TEXT CONTROL.....	18

<i>SalRTFShowToolBarButton</i>	18
TOOL BAR.....	18
<i>ToolBar Display Style</i>	18
TAB BAR.....	18
<i>SalTabUpdate</i>	18
<i>Show Close Buttons</i>	18
REPORT BUILDER.....	18
<i>Report Builder Dynamic Watermark</i>	18
<i>Report Builder Conditional Watermark</i>	19
<i>Query Conditions When Opening a CQT or Executing a CQT Query</i>	19
<i>Reports for Mobile Business Apps</i>	19
<i>RB Conditional Display Indicator</i>	19
<i>Report Builder's Clipboard</i>	19
DATABASE.....	19
<i>WIN32 TD Applications Can Use SB 11.7 Features</i>	19
DEBUGGER.....	19
<i>Removed "Show Item Information" Restriction at Debug</i>	19
<i>.NET Debugger Supports Viewing UDV's, Arrays and Arrays of UDV's</i>	20
COLOR CODING/TASKS/BOOKMARKS.....	20
<i>Logon Name Required When Creating a Task</i>	20
<i>Color-Coded Explorer Tree Nodes</i>	20
<i>Outline Color of Constants</i>	20
<i>Bookmarks in the Outline and IDE</i>	20
FILE HANDLING/LIBRARIES.....	20
<i>Default File Format</i>	20
<i>Refresh Libraries Accelerator</i>	20
<i>Simultaneously Add Multiple APL's</i>	20
<i>Images Directory</i>	20
<i>SalFileRead</i>	21
REORGANIZATION.....	21
<i>Preferences Dialog</i>	21
<i>Migration Wizard Tool</i>	21
<i>Menu Editor</i>	21
<i>Choose Outline Font Dialog</i>	21
<i>Sorting Compiler Output Messages</i>	21
<i>Class Constructors and Destructors</i>	21
ENCODING.....	21
TEAM DEVELOPER 6.2 SP3	21
RELEASE NOTES	21
SERVICE PACK PATCH INSTALLERS.....	22
SALLISTSETDROPDOWNWIDTH.....	22
TEAM DEVELOPER 6.2 SP2	22
SQLDBPIPEGETERRORINFO.....	23
SQLDBPIPESETREQUESTHEADER.....	23
TEAM DEVELOPER 6.2 SP1	24
TEAM DEVELOPER 6.2	25
IDE FEATURES	25
GUI FEATURES	26
Group By Order By Group by Summary	26

Hidden vs. visible columns Row Height.....	26
.NET NEW FEATURES	27
LANGUAGE API FEATURES	28
SALDLGOPENFILEMULTI()	29
TEAM DEVELOPER 6.1 SP4	30
TEAM DEVELOPER 6.1 SP3	30
BEHAVIOURAL DIFFERENCES	31
TEAM DEVELOPER 6.1 SP2	32
TEAM DEVELOPER 6.1 SP1	32
TEAM DEVELOPER 6.1	33
IDE FEATURES	33
GUI FEATURES	34
GRID SUMMARY BAR	34
GRID ENHANCEMENTS	35
TABLE ENHANCEMENTS.....	35
TAB BAR ENHANCEMENTS	35
CQUICKGRAPH CHART CONTROL REPLACEMENT	35
SALRIBBONMAXIMIZE,SALRIBBONMINIMIZE.....	35
NAMED TOOLBARS	35
DATE PICKER ATTRIBUTE: WEEK NUMBER	36
TREE CONTROL ENHANCEMENTS.....	36
INTERNATIONALIZATION.....	36
.NET NEW FEATURES	36
UNUSED CODE/ORPHAN ANALYZER	36
WEB SERVICES	37
ENHANCED XAML SUPPORT	37
.NET EXPLORER ENHANCEMENTS.....	37
.NET MISCELLANEOUS ENHANCEMENTS	37
DATABASE CONNECTIVITY DBPIPE.....	38
REPORT BUILDER INCLUDE SYSTEM FORMULAS	38
TEAM DEVELOPER 6.0	38
BUILD .NET APPLICATIONS	38
ENHANCED GROUP BOX CONTROL	38
ENHANCED TAB BAR CONTROL	38
NAVIGATION BAR.....	39
OUTLINE EDITOR - INSERT TASK	39
QUICKTAB2TAB CONVERSION TOOL.....	39
REPORT BUILDER "NEW QRP" OPTION	39
SALGRID EXPORT FUNCTIONS	39
SALGRIDSETALTROWBACKCOLORS	39
SALTBLCREATECOLUMNFROMCLASS	40
SALMAIL - SMTP AND POP3 SUPPORT	40
TREE CONTROL	40
<i>Note for SP3: Uninstalling TD 6.0 SP1 Full + SP3.....</i>	<i>40</i>

Team Developer 7.0

64 bit support

- Team Developer now ships with a 32 and 64 bit installer. To build 64 bit capable applications, use the 64 bit installer. The IDE will run as a 64 bit application, and any executable generated will be 64 bit capable.
- A new Sal function SallsX64 is available which will return true if the TD application is running as a 64 bit process.
- Note: In some instances, 32 bit TD applications could use incorrect data type mappings for external function parameters and return types. For instance a function with INT parameters could be defined in TD with a WORD parameter type and the function invocation would still work. This same mapping might fail in unexpected ways in a 64 bit application. Inspect your external function definitions and ensure their parameter and return types are correct.
- Note: If you're linking to TD dll's, you may have to change your compiler options. TD 7.0 (for both x86 and x64 builds) now uses default struct member alignment settings for Visual Studio (pre-7.0 versions used 1 byte alignment).

REST API

- Several new SAL functions have been added to enable consuming JSON based REST services.
 - SalJSONDeserializeUDV – Used to fetch JSON data from a REST service and deserialize the results into a UDV or UDV array.
 - SalJSONSerializeUDV – Used to serialize a UDV or UDV array into JSON, with the option of posting the data to a REST service.
- When building a web service assembly, there's a new option for enabling JSON serialization. This will allow consumers of your services to use JSON data, which can be easier to work with for some platforms.

GUI Controls

- Grid export
 - Added a SalGridDataExportEx parameter to expose additional options for exporting grids
 - EXPORT_HideHeaders – Export excludes column headers
 - EXPORT_SelectedRows – Export only selected rows
 - EXPORT_SelectedCols – Export only selected columns
 - Exporting numeric and date time columns will now create VT_R8 and COLeDateTime cells respectively
 - Excel export can now save to xlsx and template formats (xltx) by specifying those extensions in the export filename.
- Grid import
 - SalGridDataImport previously relied on an older library that is not supported on x64 platforms. The new implementation cannot open xls files generated by previous versions of TD. To work around this, you can open and re-save the .xls in Excel.
 - SalGridDataImport now supports additional file formats such as .xlsx, xltx
- Grid filter
 - Win32 grid columns have a new UI to allow users to filter data in the grid
 - New attribute on columns, "Allow Text Filtering", when enabled a filter

- icon will appear in the column header.
 - A new flag can be passed to `SalTblSetColumnFlags` to toggle this at runtime, `COL_AllowTextFilter`
- `SalGridSetSummaryColumnFmtPicture`
 - New function for setting a format picture specific to the summary field of a column.
- Status bar API
 - A new SAL API has been added to support custom status bar panes
 - `SalStatusAddPane`
 - `SalStatusSetPaneVisible`
 - `SalStatusSetPaneText`
 - `SalStatusSetPanelImage`
 - `SalStatusCreateProgressBar`
 - `SalStatusSetPaneTooltipText`
- Progress Bar
 - Added ability to display the value
 - Two new design time attributes, “Text Color” and “Show Percent” control the display of the value
 - Two new SAL API functions, `SalMeterSetTextColor` and `SalMeterSetShowPercent`
- IME (Input Method Editor) Mode
 - Data Field, Multiline Text, Combo, and grid/table columns have improved support for IME. Developers can set IME settings at design or runtime for input fields rather than requiring end users to manually enable/disable IME mode.
 - Design time attribute “IME Mode”
 - Auto: Do no change IME mode (default)
 - Active: Turn on IME mode
 - Inactive: Turn off IME mode
 - New SAL API: `SalSetIMEMode`

Language

- Function Overloading
 - You can now define functions with the same name, but different signatures.
- Protected functions/variables
 - You can now add child nodes to both functions and variables on a class to mark it as protected. Protected functions and variables can only be accessed within that class or a subclass.
- `SalWindowGetProperty` is now supported for .NET builds targets
- `SalSetLibraryPath`
 - Can be used to specify the location of an external dll to be resolved. Useful if your external assemblies are not on the system path or application folder. To use this, there’s a new compiler settings option “Disable External Function Type Checking” that must be enabled.
- `SalGetHintText/SalSetHintText` API for setting hint text for data fields, multiline text and combo box.
- `CStruct` API has been improved to use the LONG datatype internally, eliminating the 64K memory limit in previous versions of TD.

Database

- SQLBase has been upgraded to 12.0

CDK

- The CDK has been improved to add several missing types:
 - Named toolbars
 - cdkStaticWindow, cdkWindow
 - Global Resources
 - Ribbon Bar
 - Menu
 - Progress Bar
 - WPF Custom
 - Tree
 - Constructors

.NET Explorer

- Several functions/globals have been moved to the GAILBase.apl instead of being generated in the .NET wrapper APL generated by the .NET Explorer wizard. This will require that any existing wrapper APL be re-generated after migrating to TD 7 to prevent duplicate symbol errors.

Outline

- TD 7 requires older outlines to be migrated before they can be opened in the IDE. When opening outlines from previous versions of TD, a warning prompt will be displayed notifying you about the migration, with the option to cancel. During the migration, your application and its libraries will be re-saved in the new format. A backup copy will be saved as <filename>.bak.

Team Developer 6.3 SP1 Update 2

- The Report Builder Watermark dialog has been modified to include a new checkbox "Washout" when selecting a picture watermark. In previous versions of Team Developer, watermark pictures always had a washout effect applied. This new option allows this effect to be turned off by unchecking the washout checkbox. If selected, the opacity of the washout can now be controlled with the 50% and 88% radio buttons (default behavior in previous versions was fixed at 50%).

Team Developer 6.3 SP1 Update 1

- SalStrCompress and SalStrUncompress have been enhanced for .NET build targets to match the Win32 algorithm. The .NET method signature now requires a Binary datatype as .NET strings cannot hold binary data (earlier versions would base64 encode the compressed

- binary data into the receive string).
- .NET compiler now reports a compile error if a class template reference is made to an unsupported class. For instance, `SalWindowsDerivedFromClass(hWnd, cQuickTabs)` will no longer compile for .NET build targets.

Team Developer 6.3 SP1

- Added the ability to set the default font and size used by the IDE via registry settings. This is useful for improving the readability of text on high resolution displays. Under the registry key `HKEY_CURRENT_USER\SOFTWARE\Gupta\SQLWindows 6.2\Settings` the following two keys are supported:
 - `SQLWinVarPitchFont` – Can be set to a font family name.
Example: "MS Sans Serif"
 - `SQLWinVarPitchFontSize` – Can be set to a numeric value.
- `SalWindowHandleToNumber` on WPF builds has been enhanced to behave like Win32 builds. It will return a consistent value back if passed the same window handle.
- Metro theme enhancements: Border of List Box has been improved to be consistent with other controls. Combo, Grid, and Table's now have updated scrollbar style consistent with other Metro controls.
- Ribbon Bar: You can now return `FALSE` from the `SAM_RibbonShow` message to prevent the minimize/maximize of the ribbon bar by the user.
- TOM: Support for SQL Server 2014 has been added
- WPF Browser Applications: Support for command line arguments has been added. Arguments can be passed via query parameters in the URL used to access the XBAP application. For example, if an application is accessed using `sample.xbap?id=1`, then `strArgArray` will contain the passed in value.
- Three new SAL functions have been added for setting the transparent color for various GUI controls:
 - **`SalRibbonSetItemTransparentColor`**(itemOwner, itemName, clrTransparent)
 - **`SalTabSetPageTransparentColor`**(tabName, nPageIndex, clrTransparent)
 - **`SalNavSetPaneTransparentColor`**(navBarName, nPanelIndex, bExpandedIcon, clrTransparent)

Team Developer 6.3 Update 2

Team Developer 6.3 Update 2 greatly improves performance of the `VisStrSubstitute` function. This enhancement is logged as TD-22195. Gupta's

internal tests show that results can be up to 6000 times faster in Update 2 as compared to previous TD releases.

Team Developer 6.3 Update 1

Team Developer 6.3 Update 1 and future update installers will recognize whether or not a previous release of Team Developer 6.3 (including patch installs) is installed on the target machine.

A prompt to upgrade TD 6.3 will display "This setup will perform an upgrade of Team Developer 6.3. Do you want to continue?" If a release of Team Developer 6.3 is not installed on the target machine, then a full installation will occur.

Team Developer 6.3

The functional areas of TD 6.3 with new or enhanced features include:

- Reporting Features
- GUI and Control Features
- .NET Features
- IDE Features
- Language API Features
- Report Builder Features

Look and Feel

PNG and JPG Files

Portable Network Graphics (PNG) file format is now supported. The PNG format is supported in all locations where images are allowed in an application. The Joint Photographic Experts Group (JPG) file format is now supported only for use on buttons.

Web Reporting

SalWebReportCreateToFile

bOk = SalWebReportCreateToFile(sRPXFile, sOutputFile, reportVars, reportData, nOutputType, sLang)

Return value is TRUE or FALSE indicating the success or failure of the operation

Arguments:

sRPXFile: (String), the name of the RPX file which defines the report layout. This file is expected to be found on the web server and the path, if any, is relative

to the home location of the web service which is running

sOutputFile: (String), the name of the output file which will be generated. This file will be generated on the web server and its path, if any, is relative to the home location of the web service which is running

reportVars: (UDV) an instance of a class whose member fields contain any “report variables” to be used by the report engine. The names of the class fields must match the names of the report variables as defined in the RPX file.

reportData: (UDV array) an array of class objects whose member fields contain the report “data items”. The names of the class fields must match the names of the report variables as defined in the RPX file.

nOutputType: (Number) a constant specifying the kind of report file to generate:
 WRPT_OutputPDF: A PDF file
 WRPT_OutputHTML: An HTML file

sLang: (String) The language to use when generating the report (e.g. “en-US”)

Example:

Call SalWebReportCreateToFile("Invoice.rpx", "report.pdf", reportVars, inputItems, WRPT_OutputPDF)

Note that the Windows account being used by the reporting web service must have file I/O permission in order to generate these reports!

This function gives the end-user the capability of generating the reports and accumulating them on the server. At that point, the user can create a software framework, suited to their needs, for listing the available reports and returning them to the user by reading the file using SalFile* methods.

SalWebReportCreate

bOk = SalWebReportCreate(sRPXFile, bOutputFile, reportVars, reportData, nOutputType, sLang)

Return value is TRUE or FALSE indicating the success or failure of the operation.

Arguments:

sRPXFile: (String), the name of the RPX file which defines the report layout. This file is expected to be found on the web server and the path, if any, is relative to the home location of the web service which is running

bOutputFile: (Receive Binary), the generated file, either PDF or HTML, in binary format.

reportVars: (UDV) an instance of a class whose member fields contain any “report variables” to be used by the report engine. The names of the class fields must match the names of the report variables as defined in the RPX file.

reportData: (UDV array) an array of class objects whose member fields contain the report “data items”. The names of the class fields must match the names of the report variables as defined in the RPX file.

nOutputType: (Number) a constant specifying the kind of report file to generate:
 WRPT_OutputPDF: A PDF file
 WRPT_OutputHTML: An HTML file

sLang: (String) The language to use when generating the report (e.g. “en-US”)

Example:

Call `SalWebReportCreateToFile("Invoice.rpx", bMyReport, reportVars, inputItems, WRPT_OutputPDF)`

Ribbon Bar

Ribbon Gallery Control

A new ribbon bar control called a "Gallery" is added. The ribbon bar control hosts one or more images which wrap horizontally and can come in either embedded or dropdown flavors.

The Gallery control can appear in three places:

1. Inside a ribbon group. In this case, the ribbon shows as much of its contents vertically as will fit in the ribbon bar's height, and horizontally it shows as many of its Gallery Items as specified by the Columns attribute. Scroll bars allow you to scroll vertically through the rows of images and a dropdown arrow allows you to dropdown the gallery to see all the images.
2. As a Dropdown Gallery. This object can be of the Small or Large variety, similar to our button and dropdown controls. A Large-size Dropdown Gallery takes up the full height of the ribbon bar and a Small-size Dropdown Gallery takes up 1/3 of the ribbon bar height. Either way, when you click the down arrow, the Gallery is fully displayed, in a similar manner to normal galleries.
3. Within another Gallery or Dropdown Gallery. In this case, it presents as a sideways sub-menu (or dropdown within dropdown) which pops sideways to display its images.

The main content of Galleries (and Dropdown Galleries) are "Gallery Items," which are essentially clickable images.

The sizes of images contained in a gallery are taken directly from the actual sizes within the image files used. In this way, the end user can customize the sizes displayed to be almost anything

Note that Gallery controls and Dropdown Gallery controls can contain other Gallery items or simple buttons. All buttons listed will be automatically grouped together at the bottom of the Gallery control.

SalRibbonAddItem

`SalRibbonAddItem` allows the end user to dynamically create items on a ribbon bar.

`SalRibbonAddItem(Window Handle, String sName, String sParent, String sPosition, Number nRibbon_Type, Boolean bSize, String sCaption, hitem hPic)`

Window Handle: The window handle of the top-level window containing the ribbon bar

sName: The name of the new ribbon item being created

sParent: The name of the existing ribbon item which will host the new ribbon item. So if you're adding a button to a ribbon group then this would be the "name" of the group.

sPosition: The name of the existing ribbon item after which we will be adding the new ribbon item. If blank, the new ribbon item will be created at the beginning of the hosting item specified in argument 3

nRibbon_Type: One of our new RIBBON_TYPE enum values to specify the kind of item being created, e.g. Button, Dropdown, DataField, Tray.

Ribbon_Type:

1 = 'RIBBON_ITEM_TYPE_TAB'

2 = 'RIBBON_ITEM_TYPE_GROUP'

3 = 'RIBBON_ITEM_TYPE_BUTTON'

4 = 'RIBBON_ITEM_TYPE_CHECKBOX'

5 = 'RIBBON_ITEM_TYPE_COMBO'

6 = 'RIBBON_ITEM_TYPE_DATAFIELD'

7 = 'RIBBON_ITEM_TYPE_DROPDOWN'

8 = 'RIBBON_ITEM_TYPE_RADIO'

9 = 'RIBBON_ITEM_TYPE_SEPARATOR'

10 = 'RIBBON_ITEM_TYPE_TRAY'

bSize: Whether or not the control will be created in the Large-Size version.
True=Large, False=Small

sCaption: The caption, if relevant, for the new control

hPic: The name of an image resource from the Global Declarations:Resources section to use for the ribbon item

SalRibbonAddItem does not support dynamically adding ribbon galleries or drop-down galleries and their content.

SalRibbonDeleteItem

SalRibbonDeleteItem allows the end user to dynamically remove items from a ribbon bar.

SalRibbonDeleteItem(Window Handle, String)

Window Handle: The window handle of the form containing the ribbon bar

String: The name of the ribbon item to delete

SalRibbonDeleteItem does not support dynamically deleting items in a gallery or drop-down gallery.

SAM_RibbonShow

SAM_RibbonShow allows the user to control the ribbon bar being shown or hidden. The wParam will contain a 1 if the ribbon is being maximized, or expanded. The wParam will contain a 0 if the ribbon is being minimized, or contracted.

bOk = SAM_RibbonShow(1, 0)

SalRibbonPopulate

Sal_RibbonPopulate allows the user to dynamically populate ribbon bar combos from SQL at runtime.

bOk = SalRibbonPopulate(Window Handle, String, Sql Handle, String).

Returns TRUE/FALSE indicating success or failure.

Arguments:

Window Handle: A window handle to the top level window containing the ribbon bar.

String: The string reference to the combo control, of the form "tab_name.group_name.combo_name".

Sql Handle: The SQL handle to use when populating.

String: The SQL select statement to use when populating the combo.

SalRibbonSetItemImage

The user can modify ribbon item images at runtime.

bOk = SalRibbonSetItemImage(Window Handle, String, Template)

Window Handle: The form containing the ribbon bar

String: The name of the ribbon item which will receive the new image

Template: The reference to the global resource object which is the picture to insert

Returns TRUE if successful

Note that not all ribbon items have an image (e.g. tray, group). The function returns false.

SalRibbonSetItemPicture

bOk = SalRibbonSetItemPicture(hWnd, itemName, hPic)

SalRibbonSetItemPicture displays a picture in a ribbon item of type button, check button, dropdown and radio button.

Arguments:

hWnd: Window Handle. Form Window or MDI Window containing Ribbon Menu.

itemName: String. Ribbon item name.

hPic: Handle of picture to display in the ribbon item. Call VisPicLoad to create a picture handle.

Return Value:

Boolean: TRUE if successful, FALSE if an error occurs.

SaRibbonSetItemSize

bOk = SalRibbonSetItemSize(Window Handle, String, Boolean)

Window Handle: The window handle of the form containing the ribbon

String: The name of the ribbon bar item whose size is being changed

Boolean: Whether Large size is being specified. TRUE=large, FALSE=small

Note that this may cause a redesigning of the ribbon layout since small size objects combine in one column. If you switch one to being large, it'll break up the items in two columns.

Also note that not every ribbon bar item type supports 2 sizes

Return Value:

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

SalRibbonSetItemVisible

SalRibbonSetItemVisible allows the user to control the visibility of ribbon tabs and groups.

bOk = SalRibbonSetItemVisible(Window Handle, String, Boolean)

Window Handle: The window handle of the form containing the ribbon bar

String: The name of the tab or group to show/hide

Boolean: Whether to show the item or not. True = Show, False = Hide

Grid

Grid Column Summaries for Groups

The summary bar feature will be expanded to show summaries at the group level when grouping is enabled. The user can show totals and averages within each "group" of rows when grouping is turned on. When the user turns on grouping, the same statistic defined for the column for the summary bar will be shown in the footer of each group.

SalGridGetSummaryColumnValue & SalGridGetSummaryColumn

The user can return the current summary bar value for a current column without having to hand-code the logic.

bOk = SalGridGetSummaryColumnValue(Window Handle, Number, Receive Number).

Returns TRUE or FALSE indicating the success or failure of the operation.

Parameters:

Window Handle: A window handle to the grid

Number: The column identifier

Receive Number: The summary statistic for that column.

nFunction = SalGridGetSummaryColumn(Window Handle, Number)

Returns summary function type for specific column.

Parameters:

Window Handle: A window handle to the grid

Number: The column identifier

SalGridSetSummaryColumnJustify

The user can set the alignment/justification of the summary bar label and value for a column. With a specific justification on one of the columns which they use for a summary value, the user can get a summary row to reflect the same justification.

bOk = SalGridSetSummaryColumnJustify(hWndGrid, nColumnID, nJustify).

Returns boolean TRUE/FALSE indicating the success of failure of the call.

hWndGrid: Window handle, the grid containing the summary column

nColumnID: Number, the ID of the column whose summary bar value will be justified

nJustify: Number, one of our COL_Constants already in existence, to indicate the justification

CellProp_ReadOnly

The user can mark a single cell in a grid as read-only. The new nPropType constant CellProp_ReadOnly works with the existing SalGridSetCellProp function. When this property is turned on, the cell is not editable.

Popup Edit

This feature change allows the Grid Column type Popup Edit to show popup even if the editable property is set to no. The user can see the contents of non-editable popup-edit columns in grids.

CellProp_Image_HeaderFromRes & CellProp_Image_FullFromRes

The user can dynamically set the image in a grid cell by using an embedded resource. Options CellProp_Image_HeaderFromRes and CellProp_Image_FullFromRes are available for SalGridSetCellPicture. Also, option CellProp_Image_TransparentColor for SalGridSetCellProp is available to set transparent color applied to image specified by the SalGridSetCellPicture function.

SalGridDataImport

Added SalGridDataImport function.

bOk = SalGridDataImport (hWndGrid, nFile, nFileType)

Allows the user to import data from XML, CSV, text, or Excel.

Parameters

hWndGrid Window Handle. The handle (or name) of the grid.

nFile String. Name of the file to import.

nFileType Number. Use one of the following:

Format_TEXT (.txt file)

Format_XML (.xml file)

Format_EXCEL (.xls file)

Format_CSV (.csv file)

Return Value

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

Example

Pushbutton: pbCSV

Message Actions

On SAM_Click

Call SalGridDataImport(grid1,"test1.csv", Format_CSV)

SalMail

SendMailEx

With this function in the SalMail class, the user can send an email to a "blind carbon copy" (BCC) recipient.

Date Picker/Date Time Picker

SalCalGetSelectedDates

This feature allows more than one date to be selected at a time. If the user is holding down the ctrl key, he can add an additional single date to the existing selected date(s). If he holds down shift, all the dates from the last selected date through to the new one being clicked is added. If the user clicks on a date which is already selected, and the ctrl key is down, it is deselected. Deselecting a range of dates by using the shift key is not supported.

bOk = SalCalGetSelectedDates(Window Handle, Receive Date/Time array, Receive Number).

Returns true/false indicating success or failure (also false if there are no selected dates).

Arguments:

Window Handle: A window handle to the DatePicker control

Receive Date/Time array: An array of dates which are selected in the control

Receive Number: The number of dates returned

SalCalMarkDays

This Sal function allows the user to specify certain calendar days which should be visually distinct from the others on a DatePicker or DateTimePicker control.

bOk = SalCalMarkDays(Window Handle, Date/Time*, Number, Number).

Returns true/false indicating success or failure of operation.

Arguments:

Window Handle: A window handle to the DatePicker or DateTimePicker control

Date/Time Array: the array of special dates to be marked

Number: A constant specifying the type of font enhancement (May be or'd combination of FONT_EnhUnderline, FONT_EnhBold, FONT_EnhStrikeOut)

Number: A constant specifying the color (one of our COLOR_* constants).

Rich Text Control

SalRTFShowToolbarButton

The user is able to further control button visibility and enablement with these function controls: 0=disable, 1=enable, 2 = hide, and 3 = show.

Tool Bar

Toolbar Display Style

The user can hide the boarder of a toolbar with the Display Style attribute Borderless.

Tab Bar

SalTabUpdate

Syntax *bOk = SalTabUpdate (hWnd)*

Description Updates the tab bar after a SalTabAssociateChild or SalTabDisassociateChild is called.

Parameters hWnd Window Handle. The handle (or name) of the tab-bar control.

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

See Also SalTabAssociateChild and SalTabDisassociateChild

Example Actions

Call SalTabAssociateChild(TabBar1,0,dfTarget SalTabUpdate(TabBar1)

Show Close Buttons

Added "Show Close Buttons" attribute to the tab bar.

Report Builder

Report Builder Dynamic Watermark

The user can dynamically set the watermark of a report via an input variable. The user can have a watermark dynamically assigned, and not hard-coded into the QRP/CQT. When the WaterMark ribbon button in ReportBuilder is clicked, the Report Watermark dialog box opens with now a fourth radio button "Input

Variable," which has a combo box showing all of the input variables of the datatype "object". The user then can call the Sal function SalReportSetObjectVar to pass in the image to be used as the watermark.

Report Builder Conditional Watermark

The user can display watermarks, conditionally based on the value of a report "formula." This new setting will be in the Report Watermark dialog box, opened by clicking on the Watermark button on the ribbon bar (Report ribbon group). There will be a drop-down menu with the caption "Conditional Display." The dropdown will contain the names of the report formulas. If a conditional display formula is provided, the formula will be evaluated for each report page to determine whether or not that page should display the watermark.

Query Conditions When Opening a CQT or Executing a CQT Query

For Report Builder, enabled ability to leave parts of a query un-specified and then to save the needed bind information. When a CQT is opened, which has embedded SQL with dynamic binds, a dialog box opens to prompt for the bind values to be used in the SQL when it is executed.

Reports for Mobile Business Apps

Report Designer allows you to design HTML or PDF reports that enable powerful features such as formulas, multiple-break groups and easy formatting. Create invoice reports, business reports – or any type of report imaginable. At runtime, the report data is passed to the reporting engine and the completed report is opened on the client. Your reports can then be emailed to customers.

RB Conditional Display Indicator

At design time, the user can scan the layout of a report in the design pane to readily see which controls have an associated Conditional Display formula.

Report Builder's Clipboard

Report Builder's clipboard is enhanced to increase its data capacity from 64K to 1024K.

Database

WIN32 TD Applications Can Use SB 11.7 Features

Applications can now use current SB 11.7 features with the classic SQL.INI configuration.

Debugger

Removed "Show Item Information" Restriction at Debug

The IDE command "Show Item Information" is disabled while an application is being debugged.

.NET Debugger Supports Viewing UDV's, Arrays and Arrays of UDV's

The user can view the member fields of a UDV and/or the elements of an array (simple or UDV).

Color Coding/Tasks/Bookmarks

Logon Name Required When Creating a Task

In order to track more closely which users create which tasks in the outline, the user must enter When commenting as a Task we will now add the Windows logon name as a parameter

Color-Coded Explorer Tree Nodes

Objects in the explorer tree are color coded to indicate whether they come from an include file (library).

Outline Color of Constants

The user can paint constants a specific color when they appear in a line of code.

Bookmarks in the Outline and IDE

The user can choose Toggle Bookmark from the outline context menu to turn a bookmark on and off. When a bookmark is on, a flag will appear next to the node. A bookmarks dialog will allow the user to search for and visit bookmarked nodes.

File Handling/Libraries

Default File Format

Previously binary (APP), the default file type is now text (APT).

Refresh Libraries Accelerator

Added an accelerator to cause a Refresh Libraries so that user can refresh the libraries without having to go to the menu choice, Components->Refresh Libraries.

Simultaneously Add Multiple APL's

The Open dialog used by the File Include command allows the user to select more than one file (APL).

Images Directory

A directory specifically for images to be used in applications is now provided. The user can place all images for an application into a specific directory with a path to the location of the application itself. A new application property Image Path can be set in the preferences dialog on the Directories tab inside the Searching group box. The path can be either absolute or relative.

SalFileRead

Correction was made to documentation of SalFileRead function nResult. nResult is the number of characters, not the number of bytes.

Reorganization

Preferences Dialog

The “Preferences” dialog is redesigned for easier use.

Migration Wizard Tool

The “Migration Wizard” tool is redesigned for easier use.

Menu Editor

The Menu Editor has been removed.

Choose Outline Font Dialog

The “Choose Outline Font” dialog is redesigned for easier use.

Sorting Compiler Output Messages

The user can sort compiler output messages based on message type.

Class Constructors and Destructors

In supported classes, Constructor and Destructor functions are now available.

Encoding

The following pre-defined constants are defined:

- ENC_ANSI: The system default Windows ANSI code page.
- ENC_MACCP : The current system Macintosh code page.
- ENC_OEMCP : The current system OEM code page.
- ENC_UTF7 : UTF-7
- ENC_UTF8 : UTF-8

Encoding parameter can be set to the value of any code page that is installed or available in the operating system. For a list of code pages, see Code Page Identifiers ([http://msdn.microsoft.com/en-us/library/windows/desktop/dd317756\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/dd317756(v=vs.85).aspx)).

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-20929: In SalStrLop, return values that were -1 in previous versions, now correctly return 0. nCharacter is the Unicode value of the first character of strString. When strString is null, nCharacter is equal to zero (0).
- 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.

SqlDbPipeGetErrorInfo

bOk = SqlDbPipeGetErrorInfo(hSql, nError, sErrorText)

Gets the detailed DbPipe error message. This function is only valid inside a When SqlError or On SAM_SqlError error handler, and the error is 19006 (R00 DBP).

Parameters:

hSql : Sql Handle. The SQL Handle of the of the cursor that triggered the error.

nError : Receive Number. The provider specific error code (if known)

sErrorText : Receive String. The provider specific text for the error message.

Return Value:

Boolean (True if there was an error, false if not or there was an error fetching the error)

Example:

Example:

On SAM_SqlError

Call SqlExtractArgs(wParam, lParam, hSqlError, nError, nPos)

If nError = 19006

Call SqlDbPipeGetErrorInfo(hSqlError, nError, sError)

Call SalMessageBox("SqlError:" || SalNumberToStrX(nError, 0) || ": " || sError, "", 0)

Return FALSE

SqlDbPipeSetRequestHeader

bOk = SqlDbPipeSetRequestHeader (sHeaderKey, sHeaderValue,

sendSchedule, sKeyValue)

Sets an Http Request Header to the given key and value. This key and value will be sent according to the value in SendSchedule, as follows:

SCHEDULE_Always (1) - Send the key value pair indefinitely for the current session

SCHEDULE_Once (2) - Send the key value pair once.

SCHEDULE_Clear (3) - Clears a key value pair earlier set with SCHEDULE_Always, without sending any additional headers.

To allow connect, the scope of "the current session" will not be limited to the current SQL Handle, but will be global ("per process"). Note that it is the user's responsibility to ensure that the headers they specify work in their environment. Overriding headers like response-length etc. is not supported. DBPipe/win32 is implemented only for Oracle and SQLServer. It is not supported for SQLBase.

Parameters:

sHeaderKey : String. The header key to send, e.g. "Cookie"

sHeaderValue : String. The value for the header. (Note that for Cookies, the syntax for the headerValue parameter will likely need to conform to RFC 2109.)

sendSchedule : Number. SCHEDULE_Always (1), SCHEDULE_Once (2), SCHEDULE_Clear (3)

sKeyValue : Receive String. The value being sent for the given key, if any, before the call. This will only be non-null if the function was previously called with SCHEDULE_Always, otherwise it will be null.

Return Value:

Boolean (True if the new value was applied successfully, false otherwise)

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

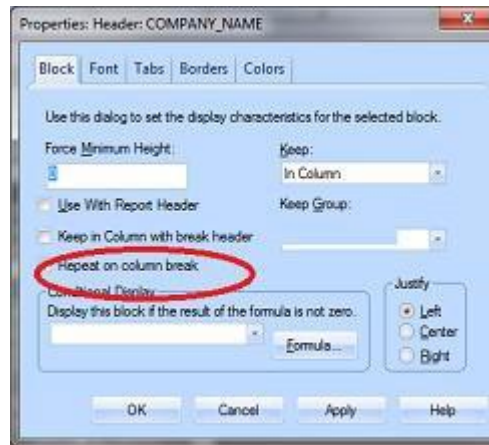
The existing support for embedding QuickGraph controls in a Report Builder report is replaced with the new Chart control. All existing QuickGraph properties are migrated forward and new Chart properties will be persisted within the QRP/CQT. Any existing QRP/CQT files will open and run in Report Builder 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 SP4

When the passed string is equal to `STRING_Null`, `SalStrLop()` should not change the value of that string and should only return zero. Otherwise the buffer length for the string would change from 2 bytes to 0 bytes, and, if you execute `SalStrLop()` on the new string, then `SalStrLop()` would return -1. Return values that were -1 in previous versions, now correctly return 0. `nCharacter` is the Unicode value of the first character of `strString`. When `strString` is null, `nCharacter` is equal to zero (0).

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_ERASEBKGD` 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 effect 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 VTTest.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 SqlErrorin` that you first specify the `When Exceptionnode`, 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 WhenException 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 SQL Windows*.

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.