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
- Enabled Breaks
- Disabled Breaks

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 incementing 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.

SalMail now supports IMAP

The API for IMAP is the same as POP3. In order to switch to IMAP, call SalMail.SetMailProtocol(MAIL_PROTOCOL_IMAP) before calling SalMail.Connect ().

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**. You can also place a label on your summary column with **SalGridSetSummaryColumnLabel**(Grid, Column Index, Label). See the in-build help for more information on these functions.

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

See **NamedToolbars.pdf** in the Books folder for more information.

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.

See **WebServicesPart1.pdf** and **WebServicesPart2.pdf** for more information.

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. To use this feature, follow these steps:

1. Within SQLWindows IDE, right click and add new Brush resource. Select XAML, right click and click on Brush Editor and select the brush type and color and then click on Ok.
2. Select any control and set the Background Brush or Foreground Brush with the resource created in Step 1.

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 occured, 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 auomatically 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."