New Features
DataWindow .NET™ 2.5

Document ID: DC00359-01-0250-01

Last revised: August 2007

New features in DataWindow .NET 2.5

This document contains information about new features added to Sybase DataWindow .NET™ since DataWindow .NET 2.0. This section contains information on the enhancements and new features in DataWindow .NET 2.5:

- DataWindow Designer Visual Studio plug-in
- TreeView Web DataWindows
- Web service as a DataWindow data source
- Support for Microsoft SQL Server 2005
- Remembering DataWindow retrieval arguments
- UseEllipsis DataWindow object property
- Changes in assembly names

For information about changes in DataWindow .NET 2.0.1, see “New features in DataWindow .NET 2.0.1” on page 7.

DataWindow Designer Visual Studio plug-in

DataWindow Designer is now provided as a set of plug-ins for Microsoft Visual Studio 2005. You can design DataWindow objects directly in Visual Studio instead of in the standalone DataWindow Designer. The standalone DataWindow Designer is also distributed with DataWindow .NET 2.5 for the convenience of customers who prefer to continue to use it.

The following modules are available as plug-ins:

- DataWindow painter

- Database painter
- Query painter
- DataWindow Project

Creating a DataWindow project

The DataWindow Project type is a new Visual Studio project type that is integrated with the Visual Studio Solution Explorer.

To create a new DataWindow project, select File>New>Project in Visual Studio .NET and select DataWindow Projects from the list of project types. A project and a library to hold your DataWindow objects are created. To create a library with a different name, right-click the new project in the Solution Explorer and select Add New Library. If you have worked with DataWindow Designer before, you can also add existing libraries to your project. Make sure that you select Migrate from the project's pop-up menu if the libraries were last used in a previous version of DataWindow .NET.

Creating a DataWindow object

To create a new DataWindow object, right-click the library where you want to save the DataWindow object in the Solution Explorer and select Add New Entry.

In the Add New Entry dialog box, select DataWindow Object from the categories list, select a DataWindow style from the Templates list, provide a name for the DataWindow object, and click Add to start the DataWindow wizard.

In the DataWindow wizard, select a data source, then complete the rest of the wizard to create the DataWindow object.

Adding controls to a DataWindow object

To add a control to a DataWindow object, select it in the DataWindow Painter category in the Visual Studio toolbox and click where you want to place it in the DataWindow painter Design view.

Setting DataWindow properties

The DataWindow Properties view has been rewritten to fit the Visual Studio look and feel. It displays in the Visual Studio Properties window. You can display properties alphabetically or by category. A brief description of each property displays at the bottom of the window if you select Description from the Properties window's pop-up menu.

For more information

For complete information about using the DataWindow Designer plug-in, including how to create projects, DataWindow objects, and queries, add controls to DataWindow objects, set properties, and open different painters and views, see the *DataWindow Designer User's Guide*.

TreeView Web DataWindows

The TreeView DataWindow style introduced in DataWindow .NET 2.0 can be used to create Web DataWindows in DataWindow .NET 2.5. The TreeView Web DataWindow supports most of the features available in the TreeView DataWindow in Windows Forms applications.

Web service as a DataWindow data source

In DataWindow .NET 2.5, you have the option of selecting a Web Service data source for all DataWindow presentation styles.

Selecting a WSDL file

When you select Web Service as the data source and click Next, the DataWindow wizard opens a page that prompts you to select a WSDL file. You can type the URL to a WSDL, ASMX, or XML file, or you can browse to a mapped drive or search a UDDI directory for these types of files.

The Choose WSDL File page of the DataWindow wizard also lets you name the assembly file that the wizard will create. The assembly file serves as an interface between the DataWindow and the Web service. If you do not name the assembly file, the wizard will select a name for you based on the name of the WSDL file entry.

Selecting a service

The next step to access a Web service data source is to select a service described in the WSDL, and then one of its public methods. You complete the wizard as you would when using any other type of data source for your DataWindow.

No SQL Select painter

After you complete the wizard, the DataWindow displays in the DataWindow painter. However, there is no equivalent to the SQL painter for a DataWindow with a Web service data source. For this type of DataWindow, you cannot select Design>Data Source from the DataWindow painter menu to change selected columns or modify the DataWindow syntax.

Runtime requirements on a deployment computer

To run the Web service DataWindow application from a deployment computer, the assembly file that you generate with the wizard must be copied along with the application executable and some required runtime DLLs. For information on the required DLLs and the Runtime Packager tool that you can use to deploy them, see “Deploying DataWindow .NET Applications” in the *Programmer’s Guide*.

For more information about using a Web service as the data source for a DataWindow, see the chapter on defining DataWindow objects in the *DataWindow Designer User's Guide*.

Using a Web service to update a database

You can use a DataWindow with a Web service data source to update a database. Support for updating data requires one or more WSDL files that describe methods and parameters that can be called by the DataWindow engine for insert, delete, or update operations.

Generating or selecting an assembly The WSDL files are not required on runtime computers. They are used to generate assembly files that are deployed with the application. If you have an existing assembly file that allows you to update data in your DataWindow objects, you can select that assembly instead of generating a new one from the Web Services Update dialog box. You can generate or select separate assemblies for insert, delete, and update operations.

Insert, delete, and update operations The insert, delete, and update operations imply different things depending on the data source. When you insert a DataWindow row for an RDBMS, a new row is added to the database; when the data source is an array of structures, a new structure instance is added to the array; and when the data source is an array of simple types, a new instance of the simple type is added to the array. The delete operation removes a database row or an instance in an array, and the update operation modifies a database row or an instance in an array.

For each operation, you must map DataWindow column values to Web service input parameters. At runtime when performing one of these operations, the DataWindow binds column values to parameters as instructed and calls the Web service method. The DataWindow engine does not know what actually happens in the Web service component (that is, how the component implements the update), only whether it returns a success or failure message.

You set values to bind to Web service parameters in the Web Service Update dialog box.

Because a DataWindow with a Web service data source does not pass back failure messages in a return code during retrieve, insert, or update operations, you must use the WebServiceException object to obtain such error information.

For more information about using a Web service to update a database, see the chapter on controlling updates in DataWindow objects in the DataWindow Designer *User's Guide*.

Support for Microsoft SQL Server 2005

A new database interface, the SNC interface, has been added to support Microsoft SQL Server. The new interface uses the SQL Server 2005 native client (*sqlncli.h* and *sqlncli.dll*) on the client side and connects using OLE DB.

PBODB initialization file not used
Connections made directly through OLE DB use the PBODB initialization file to set some parameters, but connections made using the SNC interface do not depend on the PBODB initialization file.

The SNC interface can be used to connect to SQL Server 2005 and SQL Server 2000.

For SQL Server 2000, the SQL client SDK was provided with the Microsoft Database Access Components (MDAC). MDAC does not support new features in SQL Server 2005. To use the SNC interface, the SQL Server 2005 SQL Native Client must be installed on the client computer.

For more information, see the chapter on Microsoft SQL Server in *Connecting to Your Database*.

Remembering DataWindow retrieval arguments

In previous versions of DataWindow .NET, when you specified retrieval arguments for a DataWindow object in the Specify Retrieval Arguments dialog box, the retrieval argument values that you specified were not remembered by the DataWindow object.

To enhance usability, the Specify Retrieval Arguments dialog box now has a Remember retrieval arguments check box. When the Remember retrieval arguments check box is selected, the data values are saved and the check box remains selected when you press the OK button. Clearing the Remember retrieval arguments check box and clicking OK removes the data values. This feature is only available in the development environment.

Not for nested and Composite DataWindows
You cannot use the Remember Retrieval Arguments feature in a nested or Composite DataWindow, so the Remember Retrieval Arguments check box is grayed out.

In a new DataWindow, the Remember retrieval arguments check box is disabled until you save the DataWindow and name it. You must close and reopen the DataWindow object or the Preview view to enable the checkbox.

Edit styles are created in the Object Details view

When you create or modify an edit style in the Database painter, you specify the style in the Object Details view, as you do for Display Formats and Validation Rules, instead of in a separate dialog box. You can now create an edit style for the InkEdit edit style.

UseEllipsis DataWindow object property

If a column with the Edit or EditMask edit style contains character data that is too long for the display column in the DataWindow, the data is truncated. You can choose to display an ellipsis at the end of the truncated data in Windows Forms applications. To do so, set the UseEllipsis property to True in the Behavior category in the Properties window or specify the UseEllipsis DataWindow object property in code:

```
dw1.Modify("col1.Edit.UseEllipsis=Yes")
dw1.Modify("col1.EditMask.UseEllipsis=Yes")
```

For displayed text, if the end of the string does not fit in the rectangle, it is truncated and the ellipsis is displayed. The ellipsis does not display when the column has focus.

The property is ignored if you:

- Set the HeightAutoSize property to True in the Properties window or set the Height.Autosize property in code.
- Specify an expression for the Font Escapement property in the Properties window to rotate the text or set the Font.Escapement property in code.

The UseEllipsis DataWindow object property is not supported in ASP.NET applications.

Changes in assembly names

The names of assembly files that began with *Sybase.PowerBuilder*, such as *Sybase.PowerBuilder.Db.dll*, have been changed to begin with *Sybase.DataWindow*. If you want to use a DataWindow .NET solution from a previous version with DataWindow .NET 2.5, you must replace references to these files.

For more information about migrating DataWindow .NET solutions and DataWindow Designer projects, see the DataWindow .NET 2.5 *Installation Guide*.

New features in DataWindow .NET 2.0.1

This section contains information on changes in DataWindow .NET 2.0.1:

- Showing button background color on XP
- Adaptive Server Enterprise 15 support
- Unsigned datatype support for ASE 15
- ServerCursor database parameter

Showing button background color on XP

When you change the background color of a button in a DataWindow on the XP operating system, the change does not display by default in the Preview view or when you run the application, because the XP theme typically controls the background color of buttons.

To display the change in Preview mode, set the ShowBackColorOnXP property in the General category of the Properties window for the DataWindow object.

To display the change at runtime, add the following statement to code that runs before the DataWindow is displayed:

```
dw1.Modify("DataWindow.ShowBackColorOnXP = yes")
```

Adaptive Server Enterprise 15 support

A new interface (ASE) has been added for connections to Adaptive Server® Enterprise. The ASE interface is identical to the SYC interface with two exceptions: the ASE interface supports only Adaptive Server 15 and later releases and will include support for large identifier names not available in the SYC interface. The ASE driver supports column names with up to 128 characters.

Unsigned datatype support for ASE 15

Support for three unsigned datatypes, UnsignedBigInt, UnsignedInt, and UnsignedSmallInt, introduced in Adaptive Server 15, was included in the SYC and JDBC drivers for DataWindow .NET 2.0. It is available for the ODBC and ASE drivers in DataWindow .NET 2.0.1.

ServerCursor database parameter

When you use the OLE DB database interface with a Microsoft SQL Server database and retrieve data into a DataWindow, server-side cursors are used to support multiple command execution. If this has a negative impact on performance, try increasing the size of the Block database parameter to 500 or more, or adding the following line to the [Microsoft SQL Server] section in the PBODB initialization file to turn off server-side cursors:

```
ServerCursor = 'NO'
```

The ServerCursor parameter can be used only in the PBODB initialization file.