

# Managing the Database

About this chapter

This chapter describes how you can manage a database from within Sybase WorkSpace.

Before you begin

You work with relational databases in Database Explorer. If you are not familiar with relational databases, you may want to consult an introductory text, such as *A Database Primer* by C. J. Date.

## Working with database components

A database is an electronic storage place for data. Databases are designed to ensure that data is valid and consistent and that it can be accessed, modified, and shared.

A database management system (DBMS) governs the activities of a database and enforces rules that ensure data integrity. A *relational* DBMS stores and organizes data in tables.

How you work with databases in Sybase WorkSpace

You can use Sybase WorkSpace to work with the following database components:

- Tables and columns
- Keys
- Indexes
- Database views
- Extended attributes
- Additional database components

Tables and columns

A database usually has many tables, each of which contains rows and columns of data. Each row in a table has the same columns, but a column's value for a particular row could be empty or NULL if the column's definition allows it.

Tables often have relationships with other tables. When you work with the Department table and the Employee table, the relationship between them is specified by a join of the two tables.

## Keys

Relational databases use keys to ensure database integrity.

**Primary keys** A primary key is a column or set of columns that uniquely identifies each row in a table. For example, two employees may have the same first and last names, but they have unique ID numbers. The Emp\_id column in the Employee table is the primary key column.

**Foreign keys** A foreign key is a column or set of columns that contains primary key values from another table. For example, the Dept\_id column is the primary key column in the Department table and a foreign key in the Employee table.

**Key icons** In the Database Explorer, columns defined as keys are displayed with icons. Sybase WorkSpace automatically joins tables that have a primary/foreign key relationship, with the join on the key columns

## Indexes

An index is a column or set of columns you identify to improve database performance when searching for data specified by the index. You index a column that contains information you will need frequently. Primary and foreign keys are special examples of indexes.

You specify a column or set of columns with unique values as a unique index, represented by an icon with a single key.

You specify a column or set of columns that has values that are not unique as a duplicate index, represented by an icon with two keys.

## Database views

If you often select data from the same tables and columns, you can create a database view of the tables. You give the database view a name, and each time you refer to it the associated SELECT command executes to find the data.

Database views are listed in the Objects view of the Database Explorer and can be displayed in the Object Layout view, but a database view does not physically exist in the database in the same way that a table does. Only its definition is stored in the database, and the view is re-created whenever the definition is used.

Database administrators often create database views for security purposes. For example, a database view of an Employee table that is available to users who are not in Human Resources might show all columns except Salary.

---

**Extended attributes**      Extended attributes enable you to store information about a table's columns in special system tables. Unlike tables, keys, indexes, and database views (which are DBMS-specific) extended attributes are Sybase WorkSpace-specific. The most powerful extended attributes determine the edit style, display format, and validation rules for the column.

**Additional database components**      Depending on the database to which you are connected and on your user privileges, you might be able to view or work with a variety of additional database components through Sybase WorkSpace. These components might include:

- Metadata types
- Procedures and functions
- Triggers
- Events
- Web services

You can drag most items in these folders to the Object Details view to display their properties. You can also drag procedures, functions, triggers, and events to the ISQL view.

Trigger information is listed for Adaptive Server Enterprise and Adaptive Server Anywhere (ASA) tables. A trigger is a special form of stored procedure that is associated with a specific database table. Triggers fire automatically whenever someone inserts, updates or deletes rows of the associated table. Triggers can call procedures and fire other triggers, but they have no parameters and cannot be invoked by a CALL statement. You use triggers when referential integrity and other declarative constraints are insufficient.

Events can be used in an ASA database to automate database administration tasks, such as sending a message when disk space is low. Event handlers are activated when a provided trigger condition is met. If any events are defined for an ASA connection, they display in the Events folder for the connection in the Objects view.

You can consume Web services using proxy tables in Adaptive Server databases and produce Web services using stored procedures in ASA databases.

## Managing databases

Sybase WorkSpace supports many database management systems (DBMSs). For the most part, you work the same way in Sybase WorkSpace for each DBMS, but because each DBMS provides some unique features (which Sybase WorkSpace makes use of), there are some issues that are specific to a particular DBMS.

What you can do

Using the Database Explorer, you can do the following in any DBMS to which you have been given access by the database administrator:

- Modify local table and column properties
- Retrieve, change, and insert data
- Create new local tables or modify existing tables

Changing the database connection

You can change to a different database at any time. You can have several database connections open at a time, although only one connection can be active. The database components for each open connection are listed in the Objects view.

Creating and deleting databases

Creating and deleting a database is an administrative task that you cannot do within Sybase WorkSpace.

## Using the Database Explorer Perspective

To open the Database Explorer, select Window | Open Perspective | Other | Database Explorer.

---

Views in the Database Explorer

Table 1-1 lists the views available in the Database Explorer.

**Table 1-1: Database Explorer views**

<b>View</b>	<b>Description</b>
Activity Log	Displays the SQL syntax generated by the actions you execute.
Columns	Used to create and/or modify a table's columns.
Extended Attributes	Lists the display formats, edit styles, and validation rules defined for the selected database connection.
Interactive SQL	Used to build, execute, or explain SQL.
Object Details	Displays an object's properties. For some objects, its properties are read-only; for others, properties can be modified.
Object Layout	Displays a graphical representation of tables.
Objects	Lists the database interfaces and profiles. For an open database connection, might also list all or some of the following objects associated with that database: groups, metadata types, procedures and functions, tables, columns, primary and foreign keys, indexes, users, views, driver information, events, triggers, Web services, and utilities (the database components listed depends on the database and your user privileges).
Results	Displays data in a grid, table, or freeform format.

Adding a view

If a view that you want to use does not appear in the perspective, you can add the view. Select View from the menu bar and then select the view that you want to add.

Rearranging views

To change the layout of the views in the perspective, select View | Layout | Manage from the menu bar.

Returning to the perspective's default appearance

To return the perspective to its default appearance, select View | Layout | Default from the menu bar.

Dragging and dropping

You can select certain database objects from the Objects view and drag them to the Object Details, Object Layout, Columns, and/or ISQL views. Position the pointer on the database object's icon and drag it to the appropriate view.

**Table 1-2: Using drag and drop in the Database Explorer**

Object	Can be dragged to
Driver, group, metadata type, procedure or function, table, column, user, primary or foreign key, index, event, trigger	Object Details view
Table or view	Object Layout view
Table or column	Columns view
Procedure or view	ISQL view

## Database Explorer tasks

Table 1-3 describes how to do some basic tasks in the Database Explorer. Most of these tasks begin in the Objects view. Many can be accomplished by dragging and dropping objects into different views. You can also use menu selections from pop-up menus.

**Table 1-3: Common tasks in the Database Explorer**

To do this	Do this
Modify a database profile	Highlight a database profile and select Properties from the pop-up menu.
Connect to a database	Highlight a database profile and then select Connect from the pop-up menu.
Create new profiles, tables, views, columns, keys, indexes, or groups	Highlight the database object and select New from the pop-up menu.
Modify database objects	Drag the object to the Object Details view.
Graphically display tables	Drag the table icon from the list in the Objects view to the Object Layout view, or highlight the table and select Add To Layout from the pop-up menu.
Manipulate data	Highlight the table and select Grid, Tabular, or Freeform from the pop-up menu Edit Data item.
Build, execute or explain SQL	Use the ISQL view to build SQL statements. Use the Paste Special  SQL pop-up menu to paste SELECT, INSERT, UPDATE, and DELETE statements or type them directly into the view's workspace. To execute or explain SQL, select Execute SQL or Explain SQL from the pop-up menu.
Define or modify extended attributes	Highlight the extended attribute from the list in the Extended Attributes view and select New or Properties from the pop-up menu.
Specify extended attributes for a column	Drag the column to the Object Details view and select the Extended Attributes tab.
Access database utilities	Double-click a utility in the Objects view to launch it.

To do this	Do this
Log your work	Select Design>Start Log from the Database Explorer view drop-down menu. To see the SQL syntax generated, display the Activity Log view.

## Modifying database preferences

To modify database preferences, select Design>Options from the Database Explorer view drop-down menu. Some preferences are specific to the database connection; others are specific to the Database Explorer.

Preferences on the General property page

The Use Extended Attributes, and Read Only preferences are database-connection specific preferences.

**Table 1-4: Database Explorer preferences**

Database preference	What Sybase WorkSpace does with the specified preference
Columns in the Table List	When Sybase WorkSpace displays tables graphically, eight table columns display unless you change the number of columns.
SQL Terminator Character	Sybase WorkSpace uses the semicolon as the SQL statement terminator in the ISQL Session view unless you enter a different terminator character in the box. Make sure that the character you choose is not reserved for another use by your database vendor. For example, using the slash character (/) will cause compilation errors with some DBMSs.
Refresh Table List	When Sybase WorkSpace first displays a table list, Sybase WorkSpace retrieves the table list from the database and displays it. To save time, Sybase WorkSpace saves this list internally to use again so very large table lists do not have to be regenerated. The table list is refreshed every 30 minutes (1800 seconds) unless you specify a different refresh rate.

Preferences on the Object Colors property page

You can set colors separately for each component of the Database Explorer's graphical table representation: the table header, columns, indexes, primary key, foreign keys, and joins. Set a color preference by selecting a color from a drop-down list.

You can design custom colors that you can use when you select color preferences. To design custom colors, select Design>Custom Colors from the Database Explorer view drop-down menu.

## Logging your work

As you work with your database, you generate SQL statements. As you define a new table, for example, Sybase WorkSpace builds a SQL CREATE TABLE statement internally. Similarly, when you add an index, Sybase WorkSpace builds a CREATE INDEX statement.

You can see all SQL generated in a Database Explorer session in the Activity Log view. You can also save this information to a file. This allows you to have a record of your work and makes it easy to duplicate the work if you need to create the same or similar tables in another database.

❖ **To start logging your work:**

- 1 Open the Database Explorer.
- 2 Select Design | Start Log from the Database Explorer view drop-down menu, or Start Log from the pop-up menu in the Activity Log view.

Sybase WorkSpace begins sending all generated syntax to the Activity Log view.

❖ **To stop the log:**

- Select Design | Stop Log from the Database Explorer view drop-down menu, or Stop Log from the pop-up menu in the Activity Log view.

Sybase WorkSpace stops sending the generated syntax to the Activity Log view. Your work is no longer logged.

❖ **To save the log to a permanent text file:**

- 1 Select Save or Save As from the pop-up menu in the Activity Log view.
- 2 Name the file and click Save. The default file extension is *SQL*, but you can change that if you want to.

---

### **Submitting the log to your DBMS**

You can open a saved log file and submit it to your DBMS in the ISQL view.

---

---

## Working with tables

When you open the Database Explorer, the Object view lists all tables in the current database that you have access to (including tables that were not created using Sybase WorkSpace). You can create a new table or alter an existing table. You can also modify table properties and work with indexes and keys.

### Creating a new table from scratch

In Sybase WorkSpace, you can create a new table in any database to which Sybase WorkSpace is connected.

❖ **To create a table in the current database:**

1 Do one of the following:

- Right-click in the Columns view and select New Table from the pop-up menu.
- Right-click Tables in the Objects view and select New Table from the pop-up menu.

The new table template displays in the Columns view. What you see in the view is DBMS-dependent. You use this template to specify each column in the table. The insertion point is in the Column Name box for the first column.

2 Enter the required information for this column.

As you enter information, use the Tab key to move from place to place in the column definition. After defining the last item in the column definition, press the Tab key to display the work area for the next column.

3 Repeat step 2 for each additional column in your table.

4 (Optional) Select Pending Syntax from the pop-up menu to see the pending SQL syntax.

If you have not already named the table, you must provide a name in the dialog box that displays. To hide the SQL syntax and return to the table columns, select Pending Syntax from the pop-up menu.

5 Select Save Table from the pop-up menu, then enter a name for the table in the Create New Table dialog box.

Sybase WorkSpace submits the pending SQL syntax statements it generated to the DBMS, and the table is created. The new table is displayed in the Object Layout view.

---

**About saving the table**

If you make changes after you save the table and before you close it, you see the pending changes when you select Pending SQL again. When you click Save again, Sybase WorkSpace submits a DROP TABLE statement to the DBMS, recreates the table, and applies all changes that are pending. Clicking Save many times can be time consuming when you are working with large tables, so you might want save only when you have finished.

---

- 6 Specify extended attributes for the columns.

## Creating a new table from an existing table

You can create a new table that is similar to an existing table very quickly by using the Save Table As menu option.

❖ **To create a new table from an existing table:**

- 1 Open the existing table in the Columns view by dragging and dropping it or selecting Alter Table from the pop-up menu.
- 2 Right-click in the Columns view and select Save Table As from the pop-up menu.
- 3 Enter a name for the new table and then the owner's name and click OK.  
The new table appears in the Object Layout view and the Columns view.
- 4 Make whatever changes you want to the table definition.
- 5 Save the table.
- 6 Make changes to the table's properties in the Object Details view.

## Specifying column definitions

When you create a new table, you must specify a definition for each column. The fields that display for each column in the Columns view depend on your DBMS. You may not see all of the following fields, and the values that you can enter are dependent on the DBMS.

---

For more information, see your DBMS documentation.

**Table 1-5: Defining columns in the Columns view in the Database Explorer**

Field	What you enter
Column Name	(Required) The name by which the column will be identified.
Data Type	(Required) Select a datatype from the drop-down list. All datatypes supported by the current DBMS are displayed in the list.
Width	For datatypes with variable widths, the number of characters in the field.
Dec	For numeric datatypes, the number of decimal places to display.
Null	Select Yes or No from the Null drop-down list to specify whether NULLs are allowed in the column. Specifying No means the column cannot have NULL values; users must supply a value. No is the default in a new table.
Default	The value that will be placed in a column in a row that you insert into a DataWindow object. The drop-down list has built-in choices, but you can type any other value. For an explanation of the built-in choices, see your DBMS documentation.

## Specifying table and column properties

After a table has been created and saved, you can specify the properties of a table and of any column in a table. Table properties include the fonts used for headers, labels, and data, and a comment that you can associate with the table. Column properties include the text used for headers and labels, display formats, validation rules, and edit styles used for data (also known as a column's extended attributes), and a comment you can associate with the column.

### Specifying table properties

In addition to adding a comment to associate with the table, you can choose the fonts that will be used to display information from the table in a DataWindow object. You can specify the font, point size, color, and style.

❖ **To specify table properties:**

- 1 Do one of the following:
  - Highlight the table in either the Objects view or the Object Layout view and select Properties from the pop-up menu.

- Drag and drop the table to the Object Details view.

The properties for the table display in the Object Details view.

- 2 Select a tab and specify properties:

Select this tab	To modify this property
General	Comments associated with the table
Data Font	Font for data retrieved from the database and displayed in the Results view.
Heading Font	Font for column identifiers used in grid, tabular, and n-up DataWindow objects displayed in the Results view.
Label Font	Font for column identifiers used in freeform DataWindow objects displayed in the Results view.

- 3 Right-click on the Object Details view and select Save Changes from the pop-up menu.

Any changes you made in the Object Details view are immediately saved to the table definition.

## Specifying column extended attributes

In addition to adding a comment to associate with a column, you can specify extended attributes for each column. An extended attribute is Sybase WorkSpace-specific information that enhances the definition of the column.

### ❖ To specify extended attributes:

- 1 Do one of the following:
  - Highlight the column in either the Objects view or the Object Layout view and select Properties from the pop-up menu.
  - Drag and drop the column to the Object Details view.
- 2 Select a tab and specify extended attribute values:

Select this tab	To modify these extended attributes
General	Column comments.
Headers	Label text used in freeform DataWindow objects. Header text used in tabular, grid, or n-up DataWindow objects.

Select this tab	To modify these extended attributes
Display	How the data is formatted in a DataWindow object as well as display height, width, and position. For example, you can associate a display format with a Revenue column so that its data displays with a leading dollar sign and negative numbers display in parentheses.
Validation	Criteria that a value must pass to be accepted in a DataWindow object. For example, you can associate a validation rule with a Salary column so that you can only enter a value within a particular range.  The initial value for the column. You can select a value from the drop-down list. The initial value must be the same datatype as the column, must pass validation, and can be NULL only if NULL is allowed for the column.
Edit Style	How the column is presented in a DataWindow object. For example, you can display column values as radio buttons or in a drop-down list.

- 3 Right-click on the Column property sheet and select Save Changes from the pop-up menu.

Any changes you made in the property sheet are immediately saved to the table definition.

How the information is stored

Extended attributes are stored in Sybase WorkSpace system tables in the database. Sybase WorkSpace uses the information to display, present, and validate data in the Database Explorer and in DataWindow objects. When you create a view in the Database Explorer, the extended attributes of the table columns used in the view are used by default.

About display formats, edit styles, and validation rules

In the Database Explorer, you create display formats, edit styles, and validation rules. Whatever you create is then available for use with columns in tables in the database. You can see all the display formats, edit styles, and validation rules defined for the database in the Extended Attributes view.

About headings and labels

By default, Sybase WorkSpace uses the column names as labels and headings, replacing any underscore characters with spaces and capitalizing each word in the name. For example, the default heading for the column Dept\_name is Dept Name. To define multiple-line headings, press Ctrl+Enter to begin a new line.

## Specifying additional properties for character columns

You can also set two additional properties for character columns on the Display property page: Case and Picture.

Specifying the displayed case

You can specify whether Sybase WorkSpace converts the case of characters for a column in a DataWindow object.

❖ **To specify how character data should be displayed:**

- On the Display property page, select a value in the Case drop-down list:

Value	Meaning
Any	Characters are displayed as they are entered
UPPER	Characters are converted to uppercase
lower	Characters are converted to lowercase

Specifying a column as a picture

You can specify that a character column can contain names of picture files (*BMP* or *WMF* files).

❖ **To specify that column values are names of picture files:**

- 1 On the Display property page, select the Picture check box.

When the Picture check box is selected, Sybase WorkSpace expects to find bitmap (*BMP*) or Windows metafile (*WMF*) file names in the column and displays the contents of the picture file—not the name of the file—in reports and DataWindow objects.

Because Sybase WorkSpace cannot determine the size of the image until runtime, it sets both display height and display width to 0 when you select the Picture check box.

- 2 Enter the size and the justification for the picture (optional).

## Altering a table

After a table is created, how you can alter the table depends on your DBMS.

You can always:

- Add or modify Sybase WorkSpace-specific extended attributes for columns
- Delete an index and create a new index

You can never:

- Insert a column between two existing columns
- Prohibit null values for an appended column
- Alter an existing index

---

Some DBMSs let you do the following but others do not:

- Append columns that allow null values
- Increase or decrease the number of characters allowed for data in an existing column
- Allow null values
- Prohibit null values in a column that allowed null values

---

#### **Database Explorer is DBMS aware**

The Database Explorer grays out or notifies you about actions that your DBMS prohibits.

---

For complete information about what you can and cannot do when you modify a table in your DBMS, see your DBMS documentation.

#### ❖ **To alter a table:**

- 1 Highlight the table and select Alter Table from the pop-up menu.

---

#### **Opening multiple instances of tables**

You can open another instance of a table by selecting View | Columns from the Database Explorer view drop-down menu. Doing this is helpful when you want to use the Database Explorer cut, copy, and paste features to cut or copy and paste between tables.

---

The table definition displays in the Columns view.

- 2 Make the changes you want in the Columns view or in the Object Details view.
- 3 Select Save Table or Save Changes.

Sybase WorkSpace submits the pending SQL syntax statements it generated to the DBMS, and the table is modified.

## **Cutting, copying, and pasting columns**

In the Database Explorer, you can use Cut, Copy, and Paste from the pop-up menu to cut, copy, and paste one column at a time within a table or between tables.

❖ **To cut or copy a column within a table:**

- 1 Put the insertion point anywhere in a non-editable part of the column that you want to cut or copy.
- 2 Select Cut Column or Copy Column from the pop-up menu.

❖ **To paste a column within a table:**

- 1 Put the insertion point in the column you want to paste to.  
  
If you are changing an existing table, put the insertion point in the last column of the table. If you try to insert a column between two columns, you get an error message. You can only append a column to an existing table. If you are defining a new table, you can paste a column anywhere.
- 2 Select Paste Column from the pop-up menu.

❖ **To paste a column to a different table:**

- 1 Open another instance of the Columns view and use Alter Table to display an existing table or click New to create a new table.
- 2 Put the insertion point in the column you want to paste to.
- 3 Select Paste Column from the pop-up menu.

## Closing a table

You can remove a table from a view by selecting Close or Reset View from its pop-up menu. This action only removes the table from the Database Explorer view. It does not drop (remove) the table from the database.

## Dropping a table

Dropping removes the table from the database.

❖ **To drop a table:**

- 1 Select Drop Table from the table's pop-up menu.
- 2 Click Yes.

Deleting orphaned table information

If you drop a table outside Sybase WorkSpace, information remains in the system tables about the table, including extended attributes for the columns.

---

❖ **To delete orphaned table information from the extended attribute system tables:**

- 1 Select Design>Synch Extended Attributes from the Database Explorer view drop-down menu.

If you try to delete orphaned table information and there is none, a message tells you that synchronization is not necessary.

- 2 Click Yes.

## Viewing pending SQL changes

As you create or alter a table definition, you can view the pending SQL syntax changes that will be made when you save the table definition.

❖ **To view pending SQL syntax changes:**

- Right-click the table definition in the Columns view and select Pending Syntax from the pop-up menu.

Sybase WorkSpace displays the pending changes to the table definition in SQL syntax.

The SQL statements execute only when you save the table definition or reset the view and then tell Sybase WorkSpace to save changes.

## Printing the table definition

You can print a report of the table's definition at any time, whether or not the table has been saved. The Table Definition Report contains information about the table and each column in the table, including the extended attributes for each column.

❖ **To print the table definition:**

- Select Print Definition from the pop-up menu.

## Exporting table syntax

You can export the syntax for a table to the log. This feature is useful when you want to create a backup definition of the table before you alter it or when you want to create the same table in another DBMS.

To export to another DBMS, you must have the Sybase WorkSpace interface for that DBMS.

❖ **To export the syntax of an existing table to a log:**

- 1 Select the table in the Objects or Object Layout view.
- 2 Select Export Syntax from the pop-up menu.

If you selected a table and have more than one DBMS interface installed, the DBMS dialog box displays. If you selected a view, Sybase WorkSpace immediately exports the syntax to the log.

- 3 Select the DBMS to which you want to export the syntax.
- 4 If you selected ODBC, specify a data source in the Data Sources dialog box.
- 5 Supply any information you are prompted for.

Sybase WorkSpace exports the syntax to the log. Extended attribute information (such as validation rules used) for the selected table is also exported. The syntax is in the format required by the DBMS you selected.

## About system tables

Two kinds of system tables exist in the database:

- System tables provided by your DBMS (for more information, see your DBMS documentation)
- Sybase WorkSpace extended attribute system tables

About Sybase  
WorkSpace system  
tables

Sybase WorkSpace stores extended attribute information you provide when you create or modify a table (such as the text to use for labels and headings for the columns, validation rules, display formats, and edit styles) in system tables. These system tables contain information about database tables and columns. Extended attribute information extends database definitions.

In the Employee table, for example, one column name is Emp\_lname. A label and a heading for the column are defined for Sybase WorkSpace to use in DataWindow objects. The column label is defined as Last Name:. The column heading is defined as Last Name. The label and heading are stored in the PBCatCol table in the extended attribute system tables.

The extended attribute system tables are maintained by Sybase WorkSpace. Only Sybase WorkSpace users can enter information into the extended attribute system tables. Table 1-6 lists the extended attribute system tables.

---

**Table 1-6: Extended attribute system tables**

<b>This system table</b>	<b>Stores this extended attribute information</b>
PBCatCol	Column data such as name, header and label for reports and DataWindow objects, and header and label positions
PBCatEdt	Edit style names and definitions
PBCatFmt	Display format names and definitions
PBCatTbl	Table data such as name, fonts, and comments
PBCatVld	Validation rule names and definitions

Opening and displaying system tables

You can open system tables in the Database Explorer just like other tables.

By default, Sybase WorkSpace shows only user-created tables in the Objects view. If you highlight Tables and select Show System Tables from the pop-up menu, Sybase WorkSpace also shows system tables.

## Working with keys

Why you should use keys

If your DBMS supports primary and foreign keys, you can work with the keys in Sybase WorkSpace.

If your DBMS supports them, you should use primary and foreign keys to enforce the referential integrity of your database. That way you can rely on the DBMS to make sure that only valid values are entered for certain columns instead of having to write code to enforce valid values.

For example, say you have two tables called Department and Employee. The Department table contains the column Dept\_Head\_ID, which holds the ID of the department's manager. You want to make sure that only valid employee IDs are entered in this column. The only valid values for Dept\_Head\_ID in the Department table are values for Emp\_ID in the Employee table.

To enforce this kind of relationship, you define a foreign key for Dept\_Head\_ID that points to the Employee table. With this key in place, the DBMS disallows any value for Dept\_Head\_ID that does not match an Emp\_ID in the Employee table.

For more about primary and foreign keys, consult a book about relational database design or your DBMS documentation.

What you can do in the Database Explorer

You can work with keys in the following ways:

- Look at existing primary and foreign keys

- Open all tables that depend on a particular primary key
- Open the table containing the primary key used by a particular foreign key
- Create, alter, and drop keys

For the most part, you work with keys the same way for each DBMS that supports keys. But there are some DBMS-specific issues. For complete information about using keys with your DBMS, see your DBMS documentation.

#### Viewing keys

Keys can be viewed in several ways:

- In the expanded tree view of a table in the Objects view
- As icons connected by lines to a table in the Object Layout view

---

#### **If you cannot see the lines**

If the color of your window background makes it hard to see the lines for the keys and indexes, you can set the colors for each component of the Database Explorer's graphical table representation, including keys and indexes.

---

#### Opening related tables

When working with tables containing keys, you can easily open related tables.

#### ❖ **To open the table that a particular foreign key references:**

- 1 Display the foreign key pop-up menu.
- 2 Select Open Referenced Table.

#### ❖ **To open all tables referencing a particular primary key:**

- 1 Display the primary key pop-up menu.
- 2 Select Open Dependent Table(s).

Sybase WorkSpace opens and expands all tables in the database containing foreign keys that reference the selected primary key.

---

## Defining primary keys

If your DBMS supports primary keys, you can define them in Sybase WorkSpace.

### ❖ To create a primary key:

1 Do one of the following:

- Highlight the table for which you want to create a primary key.
- Select New>Primary Key from the pop-up menu.
- Expand the table's tree view, right click Primary Key, and select New Primary Key from the pop-up menu.

The Primary Key properties display in the Object Details view.

2 Select one or more columns for the primary key.

---

#### Columns that are allowed in a primary key

Only a column that does not allow null values can be included as a column in a primary key definition. If you choose a column that allows null values, you get a DBMS error when you save the table. In DBMSs that allow rollback for Data Definition Language (DDL), the table definition is rolled back. In DBMSs that do not allow rollback for DDL, the Database Explorer is refreshed with the current definition of the table.

---

3 Specify any information required by your DBMS.

---

#### Naming a primary key

Some DBMSs allow you to name a primary key and specify whether it is clustered or not clustered. For these DBMSs, the Primary Key property page has a way to specify these properties.

---

For DBMS-specific information, see your DBMS documentation.

4 Right-click on the Object Details view and select Save Changes from the pop-up menu.

Any changes you made in the view are immediately saved to the table definition.

---

### **Completing the primary key**

Some DBMSs automatically create a unique index when you define a primary key so that you can immediately begin to add data to the table. Others require you to separately create a unique index to support the primary key before populating the table with data.

To find out what your DBMS does, see your DBMS documentation.

---

### Defining foreign keys

If your DBMS supports foreign keys, you can define them in Sybase WorkSpace.

❖ **To create a foreign key:**

1 Do one of the following:

- Highlight the table.
- Select New>Foreign Key from the pop-up menu.
- Expand the table's tree view and right click on Foreign Keys and select New Foreign Key from the pop-up menu.

The Foreign Key properties display in the Object Details view. Some of the information is DBMS-specific.

2 Name the foreign key in the Foreign Key Name box.

3 Select the columns for the foreign key.

4 On the Primary Key tab page, select the table and column containing the Primary key referenced by the foreign key you are defining.

---

### **Key definitions must match exactly**

The definition of the foreign key columns must match the primary key columns, including datatype, precision (width), and scale (decimal specification).

---

5 On the Rules tab page, specify any information required by your DBMS.

For example, you may need to specify a delete rule by selecting one of the rules listed for On Delete of Primary Table Row.

For DBMS-specific information, see your DBMS documentation.

- 
- 6 Right-click on the Object Details view and select Save Changes from the pop-up menu.

Any changes you make in the view are immediately saved to the table definition.

#### Modifying keys

You can modify a primary key in Sybase WorkSpace.

❖ **To modify a primary key:**

- 1 Do one of the following:
  - Highlight the primary key listed in the table's expanded tree view.
  - Select Properties from the pop-up menu.
  - Drag the primary key icon and drop it in the Object Details view.
- 2 Select one or more columns for the primary key.
- 3 Right-click on the Object Details view and select Save Changes from the pop-up menu.

Any changes you make in the view are immediately saved to the table definition.

#### Dropping a key

You can drop keys (remove them from the database) from within Sybase WorkSpace.

❖ **To drop a key:**

- 1 Highlight the key in the expanded tree view for the table in the Objects view or right-click the key icon for the table in the Object Layout view.
- 2 Select Drop Primary Key or Drop Foreign Key from the key's pop-up menu.
- 3 Click Yes.

## Working with indexes

You can create as many single- or multi-valued indexes for a database table as you need, and you can drop indexes that are no longer needed.

---

### **Update limitation**

You can update a table in a DataWindow object only if it has a unique index or primary key.

---

### Creating an index

---

### **In Adaptive Server Anywhere databases**

In Adaptive Server Anywhere databases, you should not define an index on a column that is defined as a foreign key, because foreign keys are already optimized for quick reference.

---

#### ❖ **To create an index:**

1 Do one of the following:

- Highlight the table for which you want to create an index.
- Select New>Index from the pop-up menu.
- Expand the table's tree view and right click on Indexes and select New Index from the pop-up menu.

The Index's properties displays in the Object Details view.

2 Enter a name for the index in the Index box.

3 Select whether or not to allow duplicate values for the index.

4 Specify any other information required for your database

For example, in Adaptive Server Enterprise specify whether the index is clustered, and in Adaptive Server Anywhere specify the order of the index.

5 Click the names of the columns that make up the index.

6 Select Save Changes from the pop-up menu.

7 Right-click on the Object Details view and select Save Changes from the pop-up menu.

Any changes you made in the view are immediately saved to the table definition.

---

Modifying an index

You can modify an index.

❖ **To modify an index:**

- 1 Do one of the following:
  - Highlight the index listed in the table's expanded tree view.
  - Select Properties from the pop-up menu.
  - Drag the index icon and drop it in the Object Details view.
- 2 In the Object Details view, select or deselect columns as needed.
- 3 Right-click on the Object Details view and select Save Changes from the pop-up menu.

Any changes you made in the view are immediately saved to the table definition.

Dropping an index

Dropping an index removes it from the database.

❖ **To drop an index from a table:**

- 1 Display the pop-up menu for the index you want to drop.
- 2 Select Drop Index and click Yes.

## Working with database views

A database view gives a different (and usually limited) perspective of the data in one or more tables. Although you see existing database views listed in the Objects view, a database view does not physically exist in the database as a table does. Each time you select a database view and use the view's data, Sybase WorkSpace executes a SQL `SELECT` statement to retrieve the data and creates the database view.

For more information about using database views, see your DBMS documentation.

Using database views  
in Sybase WorkSpace

You can define and manipulate database views in Sybase WorkSpace. Typically you use database views for the following reasons:

- To give names to frequently executed SELECT statements.
- To limit access to data in a table. For example, you can create a database view of all the columns in the Employee table except Salary. Users of the database view can see and update all information except the employee's salary.
- To combine information from multiple tables for easy access.

In Sybase WorkSpace, you can create single- or multiple-table database views. You can also use a database view when you define data to create a new database view.

You define, open, and manipulate database views in the View editor, which is similar to the SQL Dialog (also known as Visual SQL).

---

**Updating database views**

Some database views are logically updatable and others are not. Some DBMSs do not allow any updating of views. For the rules your DBMS follows, see your DBMS documentation.

---

❖ **To open a database view:**

- 1 In the Objects view, expand the list of Views for your database.
- 2 Highlight the view you want to open and select Add To Layout from the pop-up menu, or drag the view's icon to the Object Layout view.

❖ **To create a database view:**

- 1 Select New View from the pop-up menu.

The Select Tables dialog box displays listing all tables and views that you can access in the database.

- 2 Select the tables and views from which you will create the view by doing one of the following:
  - Click the name of each table or view you want to open in the list displayed in the Select Tables dialog box, then click the Open button to open them. The Select Tables dialog box closes.
  - Double-click the name of each table or view you want to open. Each object is opened immediately. Then click the Cancel button to close the Select Tables dialog box.

---

Representations of the selected tables and views display in the View editor workspace.

- 3 Select the columns to include in the view and include computed columns as needed.
- 4 Join the tables if there is more than one table in the view.
- 5 Specify criteria to limit rows retrieved (Where tab), group retrieved rows (Group tab), and limit the retrieved groups (Having tab) if appropriate.
- 6 When the view has been completed, click the OK button.
- 7 Name the view.

Include *view* or some other identifier in the view's name so that you will be able to distinguish it from a table in the Select Tables dialog box.

- 8 Click the Create button.

Sybase WorkSpace generates a CREATE VIEW statement and submits it to the DBMS. The view definition is created in the database. You return to the Database Explorer workspace with the new view displayed in the workspace.

#### Displaying a database view's SQL statement

You can display the SQL statement that defines a database view. How you do it depends on whether you are creating a new view in the View editor or want to look at the definition of an existing view.

❖ **To display the SQL statement from the View painter:**

- Select the Preview tab in the View editor.

Sybase WorkSpace displays the SQL it is generating. The display is updated each time you change the view.

❖ **To display the SQL statement from the Database Explorer:**

- Highlight the name of the database view in the Objects view and select Properties from the pop-up menu, or drag the view's icon to the Object Details view.

The completed SELECT statement used to create the database view displays in the Definition field on the General page.

---

**View dialog box is read-only**

You cannot alter the view definition in the Object Details view. To alter a view, drop it and create another view.

---

**Joining tables** If the database view contains more than one table, you should join the tables on their common columns. When the View editor is first opened for a database view containing more than one table, Sybase WorkSpace makes its best guess as to the join columns, as follows:

- If there is a primary/foreign key relationship between the tables, Sybase WorkSpace automatically joins them.
- If there are no keys, Sybase WorkSpace tries to join tables based on common column names and types.

❖ **To join tables:**

- 1 Click the Join button.
- 2 Click the columns on which you want to join the tables.
- 3 To create a join other than the equality join, click the join representation in the workspace.

The Join dialog box displays.

- 4 Select the join operator you want from the Join dialog box.

If your DBMS supports outer joins, outer join options also display in the Join dialog box. For example, if joining the Employee and Department tables, you can choose to include rows from the Employee table where there are no matching departments, or rows from the Department table where there are no matching employees.

**Dropping a database view** Dropping a database view removes its definition from the database.

❖ **To drop a view:**

- 1 In the Objects view, select the database view you want to drop.
- 2 Select Drop View from the pop-up menu.

Sybase WorkSpace prompts you to confirm the drop, then generates a DROP VIEW statement and submits it to the DBMS.

**Exporting view syntax** You can export the syntax for a view to the log. This feature is useful when you want to create a backup definition of the view before you alter it or when you want to create the same view in another DBMS.

❖ **To export the syntax of an existing view to a log:**

- 1 Select the view in the painter workspace.
- 2 Select Export Syntax from the pop-up menu.

---

## Manipulating data

As you work on the database, you often want to look at existing data or create some data for testing purposes. You might also want to test display formats, validation rules, and edit styles on real data.

Sybase WorkSpace provides data manipulation for such purposes. With data manipulation, you can:

- Retrieve and manipulate database information
- Save the contents of the database in a variety of formats (such as Excel, PDF, or XML)

## Retrieving data

### ❖ To retrieve data:

- 1 In the Database Explorer, select the table or database view whose data you want to manipulate.
- 2 Select Edit Data from the pop-up menu and choose one of the edit options from the cascading menu that displays.

All rows are retrieved and display in the Results view. As the rows are being retrieved, a red button is displayed in the status bar. You can click the button to stop the retrieval.

Exactly what you see in the Results view depends on the formatting style you picked. What you are seeing is actually a DataWindow object. The formatting style you picked corresponds to a type of DataWindow object (grid, tabular, or freeform). In a grid display, you can drag the mouse on a column's border to resize the column.

Only a few rows of data display at a time. You can select First, Prior, Next, or Last from the pop-up menu to move from page to page.

## Modifying data

You can add, modify, or delete rows. When you have finished manipulating the data, you can apply the changes to the database.

---

**If looking at data from a view**

Some views are logically updatable and others are not. Some DBMSs do not allow any updating of views.

For the rules your DBMS follows regarding updating of views, see your DBMS documentation.

---

❖ **To modify data:**

1 Do one of the following:

- To modify existing data, tab to a field and enter a new value.
- To add a row, select Insert Row from the pop-up menu and enter data in the new row.
- To delete a row, select Delete Row from the pop-up menu.

When you add or modify data, the data uses the validation rules, display formats, and edit styles that you or others have defined for the table in the Database Explorer.

2 Select Save from the pop-up menu to apply changes to the database.

## Saving data

You can save the displayed data in an external file.

❖ **To save the data in an external file:**

1 Select Save Rows As from the pop-up menu.

The Save Rows As dialog box displays.

2 Choose a format for the file.

You can select from several formats, including Powersoft report (PSR), XML, and HTML.

If you want the column headers saved in the file, select a file format that includes headers, such as Excel With Headers. When you select a *with headers* format, the names of the database columns (not the column labels) will also be saved in the file.

3 For TEXT, CSV, SQL, HTML, and DIF formats, select an encoding for the file.

---

You can select ANSI/DBCS, Unicode LE (Little-Endian), Unicode BE (Big-Endian), or UTF8.

- 4 Name the file and save it.

Sybase WorkSpace saves all displayed rows in the file; all columns in the displayed rows are saved.

## Creating and executing SQL statements

The Database Explorer's Interactive SQL view is a SQL editor in which you can enter and execute SQL statements. The view provides all editing capabilities needed for writing and modifying SQL statements. You can cut, copy, and paste text; search for and replace text; and create SQL statements. You can also set editing properties to make reading your SQL files easier.

## Building and executing SQL statements

You can use the Interactive SQL view to build SQL statements and execute them immediately. The view acts as a notepad in which you can enter SQL statements.

## Creating stored procedures

You can use the Interactive SQL view to create stored procedures or triggers, but make sure that the Database Explorer's SQL statement terminator character is not the same as the terminator character used in the stored procedure language of your DBMS.

### About the statement terminator

By default, Sybase WorkSpace uses the semicolon as the SQL statement terminator. You can override the semicolon by specifying a different terminator character in the Database Explorer. To change the terminator character, select Design>Options from the Database Explorer's view drop-down menu

Make sure that the character you choose is not reserved for another use by your database vendor. For example, using the slash character (/) causes compilation errors with some DBMSs.

## Controlling comments

By default, Sybase WorkSpace strips off comments when it sends SQL to the DBMS. You can have comments included by clearing the check mark next to Strip Comments in the pop-up menu of the Interactive SQL view.

## Entering SQL

You can enter a SQL statement in four ways:

- Pasting the statement
- Typing the statement in the view
- Opening a text file containing the SQL
- Dragging a procedure or function from the Objects view

### Pasting SQL

You can paste SELECT, INSERT, UPDATE, and DELETE statements to the view. Depending on which kind of statement you want to paste, Sybase WorkSpace displays dialog boxes that guide you through creating the full statement.

#### ❖ To paste a SQL statement to the workspace:

- 1 Select Paste Special>SQL from the pop-up menu, then the statement type (Select, Insert, Update, or Delete).

The Select Table dialog box displays.

- 2 Select the table(s) you will reference in the SQL statement.

Go to the SQL Dialog, to create a Select, Insert, Update, or Delete statement, depending on the statement type you selected.

- 3 Do one of the following:

- For a SELECT statement, define the statement in the same way as when building a database view.  
  
You choose the columns to select. You can define computed columns, specify sorting and joining criteria, and WHERE, GROUP BY, and HAVING criteria. For more information, see “Working with database views” on page 25.
- For an INSERT statement, type the values to insert into each column. You can insert as many rows as you want.
- For an UPDATE statement, specify the new values for the columns in the Update Column Values dialog box. Then specify the WHERE criteria to indicate which rows to update.

- 
- For a DELETE statement, specify the WHERE criteria to indicate which rows to delete.

4 When you complete creating the SQL statement, click OK.

You return to the Database Explorer with the SQL statement pasted into the ISQL view.

#### Typing SQL

Rather than paste, you can simply type one or more SQL statements directly in the ISQL view.

You can enter most statements supported by your DBMS. You cannot enter certain statements that could destabilize the Sybase WorkSpace development environment. These include the SET statement and the USE *database* statement.

---

#### **Sybase Adaptive Server Enterprise stored procedures**

When you use the Database Explorer to execute a Sybase Adaptive Server Enterprise system stored procedure, you *must* start the syntax with the keyword EXEC or EXECUTE. For example, enter EXEC SP\_LOCK. You cannot execute the stored procedure simply by entering its name.

---

#### Importing SQL from a text file

You can import SQL that has been saved in a text file into the Database Explorer.

❖ **To read SQL from a file:**

- 1 Put the insertion point where you want to insert the SQL.
- 2 Select Paste Special>From File from the pop-up menu.
- 3 Select the file containing the SQL and click OK.

#### Dragging a procedure or function from the Objects view

From the tree view in the Objects view, you can select an existing procedure or function that contains a SQL statement you want to enter and drag it to the Interactive SQL view.

## Explaining SQL

Sometimes there is more than one way to code SQL statements to obtain the results you want. When this is the case, you can use Explain SQL on the pop-up menu to help you select the most efficient method. Explain SQL displays information about the path that Sybase WorkSpace will use to execute the statements in the SQL Statement Execution Plan dialog box. This is most useful when you are retrieving or updating data in an indexed column or using multiple tables.

---

### **DBMS-specific information**

The information displayed in the SQL Statement Execution Plan dialog box depends on your DBMS. For more about the SQL execution plan, see your DBMS documentation.

---

## **Executing SQL**

When you have the SQL statements you want in the workspace, you can submit them to the DBMS.

❖ **To execute the SQL:**

- Select Execute... from the pop-up menu.

If the SQL retrieves data, the data appears in grid format in the Results view. If there is a database error, you see a message box describing the problem.

For a description of what you can do with the data, see “Manipulating data” on page 29.

## **Customizing the editor**

The Interactive SQL view has Script, Font, and Coloring properties that you can change to make SQL files easier to read. With no change in properties, SQL files have black text on a white background and a tab stop setting of 3 for indentation.

Setting Script and  
Font properties

Select Design>Options from the Database Explorer view drop-down menu to open the Database Preferences dialog box. You can set Script and Font properties in the Script and Editor Font tab pages.

Setting Coloring  
properties

You can set the text color and background color for SQL styles (such as datatypes and keywords) so that the style will stand out and the SQL code will be more readable. You set Coloring properties on the Coloring tab page.

---

### **Enabling syntax coloring**

Be sure the Enable Syntax Coloring check box is selected before you set colors for SQL styles. You can turn off all Coloring properties by clearing the check box.

---

---

## Consuming and producing Web services

A Web service is a self-contained, modular application that can be accessed through a network connection. Web services provide interoperability on multiple platforms by enforcing adherence to the Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), HTML, and Extensible Markup Language (XML) open standards.

### Consuming Web services in Sybase WorkSpace

In the Database Explorer

In the Database Explorer, you can map a Web service definition to one or more proxy tables in an Adaptive Server Enterprise database, then use the proxy tables to define queries and SELECT statements. You can also use proxy tables that have already been defined in the database.

This feature uses the Web Services Consumer component in Adaptive Server 12.5.2 and later. For more information, see the *Web Services User's Guide* in the Adaptive Server 12.5.2 documentation set on the Sybase Product Manuals Web site at <http://sybooks.sybase.com/onlinebooks/group-as/asg1252e/wbsrvcs>.

### Web Services as Tables folder

The Web Services as Tables folder at the top level of an Adaptive Server Enterprise connection profile in the Objects view displays all proxy tables that have been mapped from Web service definitions in the current database. From its pop-up menu, you can map a new Web service to one or more proxy tables and refresh the view of proxy tables in the database.

### Mapping a Web service definition to a proxy table

To create a new Web service proxy table, select Map Web Service to Tables from the pop-up menu for the Web Services As Tables folder in an Adaptive Server Enterprise connection in the Objects view. The Select WSDL dialog is opened.

In the dialog, you can enter or browse for the name or the URL of a WSDL file, or you can search for a Web service using Universal Description, Discovery, and Integration (UDDI) registries. The UDDI service is an industry-wide effort to bring a common standard for business-to-business integration. It defines a set of standard interfaces for accessing a database of Web services.

## Using the UDDI

To use a UDDI search, click the Browse UDDI... button, which opens the Sybase WorkSpace UDDI Inquiry Wizard.

When you complete the wizard, the Web Services Consumer component maps the Web methods in the WSDL file to tables. The component derives the names of tables and columns from the names of the Web methods.

Each proxy table contains a Columns folder, which in turn contains column objects that represent the input and output parameters of the Web service method that corresponds to the proxy table. All input column names start with an underscore character. For document/literal methods, the proxy table has two methods: `_inxml` and `outxml`.

The new proxy tables display in the Web Services as Tables folder in the Objects view, and in the Select Tables dialog box in the SQL Query editor and SQL Dialog.

## Working with proxy tables

You can view a proxy table's properties in the Object Details view.

There are some restrictions on the use of proxy tables in defining queries and SQL statements. For example, all input columns must be used to define a WHERE clause. If a Web service has an input method that maps to a `_zipcode` column in a proxy table named `gettemp`, and an output method that maps to a `retval` column, the `_zipcode` input column must appear in a WHERE clause. A valid SELECT statement might be:

```
SELECT retval FROM gettemp WHERE _zipcode='01730'
```

Neither the SQL Query editor nor the SQL Dialog enforces any restrictions or limitations on the use of proxy tables in SELECT, INSERT, UPDATE, or other SQL statements. The validity of the statement is not tested until it is executed. For a complete description of the restrictions that apply to the use of proxy tables, see the *Web Services User's Guide* in the Adaptive Server 12.5.2 documentation set on the Sybase Product Manuals Web site at <http://sybooks.sybase.com/onlinebooks/group-as/asg1252e/wbsrvcs>

## Producing Web services in Sybase WorkSpace

### In the Database Explorer

In the Database Explorer, you can create a SOAP Web service based on an ASA stored procedure and save it to the database, as well as optionally create a SOAP service model.

---

## ASA Web Services folder

The ASA Web Services folder at the top level of an ASA connection profile in the Objects view displays all SOAP Web services that have been created from ASA stored procedures in the current database. From its pop-up menu, you can create a new Web service and refresh the view of Web services in the database.

## Creating a Web service from an ASA stored procedure

You can create a Web service from the pop-up menu for a stored procedure in the Procedures and Functions folder, or from the pop-up menu for the ASA Web Services folder.

❖ **To create a SOAP Web service from an ASA stored procedure:**

- 1 Select Create Web Service from the pop-up menu for a stored procedure or the Web Services folder.  
If you are in the Database Development perspective, you are notified you will be switched to the Database Explorer perspective.
- 2 Set the properties listed in the Object Details view.
- 3 If Create SOAP Service is checked, after the ASA Web service is created, a SOAP service model will be created with the new Web service as the endpoint.

Clear the checkbox if you do not want to create a SOAP service model.

- 4 Select Save Changes from the pop-up menu in the Object Details view.

The Web service is added to the Web Services folder in the Objects view.

If Create SOAP Service was checked, specify a name and location for the service model.

If Authorization was checked, specify a user id and password.

The SOAP service model is added to the specified location and displayed in the WorkSpace Navigator.

**Table 1-7: ASA Web service properties**

Property	Description
Web Service Name	The name of the Web service. The name cannot be modified after the Web service is saved. The default is the name of the stored procedure with the prefix ws_.
Stored Procedure	The name of the stored procedure. If you selected Create Web Service from the pop-up menu for a stored procedure, its name is displayed by a default.

Property	Description
Authorization	Select if users must provide a user name and password to use the Web service: <ul style="list-style-type: none"><li>To specify that a specific user name and password must be supplied, select Authorization and Run As User and select a user name from the User drop-down list.</li><li>To specify that any user known to the database can use the Web service, select Authorization and clear the Run As User check box.</li><li>To specify that a user name and password are not required, clear the Authorization check box. Clearing the Authorization check box disables the Run As User check box and sets it to checked. All requests are run using the account of the user specified in the User drop-down list.</li></ul>
Run As User	Select if you have selected Authorization and want to specify that only the user selected from the User drop-down list can use the Web service. Clear if any user can use the Web service.
User	Select a user from the drop-down list. The default is DBA.
Secure	Select if security is required for the Web service. When this box is selected, the Web service is processed only if the request is made using a secure (https) connection. If a request is received on the http port, the request is redirected to the https port.

## Using an ASA Web service

An ASA Web service can be used as an endpoint for a SOAP service or accessed in a browser by using the following syntax (use https for a secure service):

```
http://host_name:port/server_name/service_name
```

For example, if the name of the stored procedure is `sp_test` and you accept the default service name of `ws_sp_test`, the following URL invokes the service on an ASA Web server called `MyWebServer` that is running on a host named `apollo` and is listening on the port 80:

```
http://apollo:80/MyWebServer/ws_sp_test
```

---

## Working with stored procedures that define Web services

When you look at the properties of a stored procedure that has been used to define Web services, a read-only list of the Web services based on the stored procedure displays in the Properties view.

If you attempt to drop a stored procedure that has been used to define a Web service, you will see a warning that the Web services exist and you can choose to confirm or cancel the drop. Dropping the procedure removes the Web services.

