



Web Services User's Guide

Adaptive Server® Enterprise

12.5.2

DOCUMENT ID: DC10061-01-1252-01

LAST REVISED: April 2004

Copyright © 1989-2004 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor. Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase, the Sybase logo, AccelaTrade, ADA Workbench, Adaptable Windowing Environment, Adaptive Component Architecture, Adaptive Server, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Server Enterprise Monitor, Adaptive Server Enterprise Replication, Adaptive Server Everywhere, Adaptive Server IQ, Adaptive Warehouse, Anywhere Studio, Application Manager, AppModeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-Translator, APT-Library, AvantGo, AvantGo Application Alerts, AvantGo Mobile Delivery, AvantGo Mobile Document Viewer, AvantGo Mobile Inspection, AvantGo Mobile Marketing Channel, AvantGo Mobile Pharma, AvantGo Mobile Sales, AvantGo Pylon, AvantGo Pylon Application Server, AvantGo Pylon Conduit, AvantGo Pylon PIM Server, AvantGo Pylon Pro, Backup Server, BizTracker, ClearConnect, Client-Library, Client Services, Convoy/DM, Copernicus, Data Pipeline, Data Workbench, DataArchitect, Database Analyzer, DataExpress, DataServer, DataWindow, DB-Library, dbQueue, Developers Workbench, Direct Connect Anywhere, DirectConnect, Distribution Director, e-ADK, E-Anywhere, e-Biz Impact, e-Biz Integrator, E-Whatever, EC Gateway, ECMAP, ECRTP, eFulfillment Accelerator, Embedded SQL, EMS, Enterprise Application Studio, Enterprise Client/Server, Enterprise Connect, Enterprise Data Studio, Enterprise Manager, Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, eProcurement Accelerator, EWA, Financial Fusion, Financial Fusion Server, Gateway Manager, GlobalFIX, iAnywhere, ImpactNow, Industry Warehouse Studio, InfoMaker, Information Anywhere, Information Everywhere, InformationConnect, InternetBuilder, iScript, Jaguar CTS, jConnect for JDBC, Mail Anywhere Studio, MainframeConnect, Maintenance Express, Manage Anywhere Studio, M-Business Channel, M-Business Network, M-Business Server, MDI Access Server, MDI Database Gateway, media.splash, MetaWorks, My AvantGo, My AvantGo Media Channel, My AvantGo Mobile Marketing, MySupport, Net-Gateway, Net-Library, New Era of Networks, ObjectConnect, ObjectCycle, OmniConnect, OmniSQL Access Module, OmniSQL Toolkit, Open Biz, Open Client, Open ClientConnect, Open Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, Orchestration Studio, PB-Gen, PC APT Execute, PC DB-Net, PC Net Library, PocketBuilder, Pocket PowerBuilder, Power++, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, PowerJ, PowerScript, PowerSite, PowerSocket, Powersoft, PowerStage, PowerStudio, PowerTips, Powersoft Portfolio, Powersoft Professional, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, Rapport, RepConnector, Replication Agent, Replication Driver, Replication Server, Replication Server Manager, Replication Toolkit, Report-Execute, Report Workbench, Resource Manager, RW-DisplayLib, RW-Library, S-Designer, SDF, Secure SQL Server, Secure SQL Toolset, Security Guardian, SKILS, smart.partners, smart.parts, smart.script, SQL Advantage, SQL Anywhere, SQL Anywhere Studio, SQL Code Checker, SQL Debug, SQL Edit, SQL Edit/TPU, SQL Everywhere, SQL Modeler, SQL Remote, SQL Server, SQL Server Manager, SQL SMART, SQL Toolset, SQL Server/CFT, SQL Server/DBM, SQL Server SNMP SubAgent, SQL Station, SQLJ, STEP, SupportNow, S.W.I.F.T. Message Format Libraries, Sybase Central, Sybase Client/Server Interfaces, Sybase Financial Server, Sybase Gateways, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase User Workbench, SybaseWare, Syber Financial, SyberAssist, SyBooks, System 10, System 11, System XI (logo), SystemTools, Tabular Data Stream, TotalFix, TradeForce, Transact-SQL, Translation Toolkit, UltraLite, UltraLite.NET, UNIBOM, Unilib, Uninull, Unisep, Unistring, URK Runtime Kit for UniCode, VisualWriter, VQL, WarehouseArchitect, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, Watcom, Watcom SQL, Watcom SQL Server, Web Deployment Kit, Web.PB, Web.SQL, WebSights, WebViewer, WorkGroup SQL Server, XA-Library, XA-Server and XP Server are trademarks of Sybase, Inc. 02/04

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names used herein may be trademarks or registered trademarks of their respective companies.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

About This Book	vii
-----------------------	-----

PART 1 INTRODUCTION TO WEB SERVICES

CHAPTER 1	Understanding Adaptive Server Enterprise Web Services	3
	Overview	3
	Components of Adaptive Server Enterprise Web Services.....	4
	Advantages of Adaptive Server Enterprise Web Services	4
	Stored procedures and functions	5
	SQL	5
	Security	5
	LDAP	5
	Web services standards.....	6
	XML.....	6
	WSDL	9
	SOAP	10
CHAPTER 2	Understanding the Web Services Producer	15
	Overview	15
	Components	16
	HTTP handler	16
	SOAP handler	16
	XML mapper.....	16
CHAPTER 3	Understanding the Web Services Consumer	17
	Overview	17
	Components.....	18
	SDS server	18
	SOAP stack	18

PART 2 CONFIGURATION

CHAPTER 4	Configuring Adaptive Server Enterprise Web Services.....	23
	Configuration	23
	Configuring during installation	24
	Configuring after installation	24
	Licensing	25
	Configuration files	26
	ws.properties	26
	logging.properties	26
	wsmsg.properties	26
	Security	26
	Configuring SSL	27
	Installing a certificate for Microsoft .NET	28
PART 3	ADMINISTRATION	
CHAPTER 5	Using Adaptive Server Enterprise Web Services	33
	Using the Web Services Producer	33
	Starting the Web Services Producer	33
	Stopping the Web Services Producer	36
	Adaptive Server Enterprise Web Services methods	38
	Using the Web Services Consumer	42
	Starting the Web Services Consumer	42
	Stopping the Web Services Consumer	44
	Using sp_webservices.....	46
	Invoking a Web service	49
	Adaptive Server Enterprise Web Services Logging	54
	Adaptive Server Enterprise Web Services log files	55
	Rolling over log files	55
	Sybase Central.....	55
	Using the Web interface	56
	Performing administrative tasks	57
	Executing Web methods	57
CHAPTER 6	Producer Sample Applications.....	59
	Apache sample client	59
	Creating the sample client.....	59
	runexecute.....	60
	Microsoft .NET sample client	64
	Creating the sample client.....	64
	Execute.exe.....	64

PART 4	TROUBLESHOOTING	
CHAPTER 7	Troubleshooting.....	69
	Troubleshooting issues	69
	HTTP-authorized Web methods unusable	69
	Encoding issue for roman8 character set on HP	69
	Unmapped RPC/encoded Web method	70
	Truncated document/literal results	70
	Starting the Producer and Consumer	70
	Locating WSDL	71
	Specifying entries in ws.properties	71
	Windows NT command-line arguments	71
	Run or stop scripts fail	72
	runexecute or execute fails	72
	Null passwords	73
	Specifying SOAP endpoints with SSL	73
	Abnormal termination of sp_webservices 'add'	73
	Web Services proxy table restrictions	74
	sysattributes table entry	75
	Diagnostic tools	75
	Enabling JDBC-level tracing	75
	Messages	75
PART 5	APPENDIXES	
APPENDIX A	Installation Contents	81
	Adaptive Server Enterprise Web Services directory tree	81
	bin directory contents	81
	lib directory contents	83
	logs directory contents	83
	producer directory contents	84
	props directory contents	84
	samples directory contents	84
APPENDIX B	Configuration Properties.....	87
	ws.properties	87
	myres.properties	89
APPENDIX C	SOAP and Adaptive Server Enterprise Datatype Mapping	91
	Datatype mapping	91

Contents

Glossary	95
Index	97

About This Book

Audience

This document is intended for users interested in accessing the Web services provided by Adaptive Server® Enterprise or in using Adaptive Server Enterprise to access the Web services of other applications.

How to use this book

This document consists of the following parts:

- Part 1, “Introduction to Web Services”
 - Chapter 1, “Understanding Adaptive Server Enterprise Web Services”
 - Chapter 2, “Understanding the Web Services Producer”
 - Chapter 3, “Understanding the Web Services Consumer”
- Part 2, “Configuration”
 - Chapter 4, “Configuring Adaptive Server Enterprise Web Services”
- Part 3, “Administration”
 - Chapter 5, “Using Adaptive Server Enterprise Web Services”
 - Chapter 6, “Producer Sample Applications”
- Part 4, “Troubleshooting”
 - Chapter 7, “Troubleshooting”
- Part 5, “Appendixes”
 - Appendix A, “Installation Contents”
 - Appendix B, “Configuration Properties”
 - Appendix C, “SOAP and Adaptive Server Enterprise Datatype Mapping”

Related documents

The Adaptive Server documentation set consists of the following:

- The release bulletin for your platform – contains last-minute information that was too late to be included in the books.

A more recent version of the release bulletin may be available on the World Wide Web. To check for critical product or document information that was added after the release of the product CD, use the Sybase® Technical Library.

- *The Installation Guide* for your platform – describes installation, upgrade, and configuration procedures for all Adaptive Server and related Sybase products.
- *What's New in Adaptive Server Enterprise?* – describes the new features in Adaptive Server version 12.5.1, the system changes added to support those features, and the changes that may affect your existing applications.
- *ASE Replicator User's Guide* – describes how to use the Replicator feature of Adaptive Server to implement basic replication from a primary server to one or more remote Adaptive Servers.
- *Component Integration Services User's Guide* – explains how to use the Adaptive Server Component Integration Services feature to connect remote Sybase and non-Sybase databases.
- *Configuring Adaptive Server Enterprise* for your platform – provides instructions for performing specific configuration tasks for Adaptive Server.
- *EJB Server User's Guide* – explains how to use EJB Server to deploy and execute Enterprise JavaBeans in Adaptive Server.
- *Error Messages and Troubleshooting Guide* – explains how to resolve frequently occurring error messages and describes solutions to system problems frequently encountered by users.
- *Full-Text Search Specialty Data Store User's Guide* – describes how to use the Full-Text Search feature with Verity to search Adaptive Server Enterprise data.
- *Glossary* – defines technical terms used in the Adaptive Server documentation.
- *Historical Server User's Guide* – describes how to use Historical Server to obtain performance information for SQL Server® and Adaptive Server.
- *Java in Adaptive Server Enterprise* – describes how to install and use Java classes as datatypes, functions, and stored procedures in the Adaptive Server database.

- *Job Scheduler User's Guide* – provides instructions on how to install and configure, and create and schedule jobs on a local or remote Adaptive Server using the command line or a graphical user interface (GUI).
- *Monitor Client Library Programmer's Guide* – describes how to write Monitor Client Library applications that access Adaptive Server performance data.
- *Monitor Server User's Guide* – describes how to use Monitor Server to obtain performance statistics from SQL Server and Adaptive Server.
- *Performance and Tuning Guide* – is a series of four books that explains how to tune Adaptive Server for maximum performance:
 - *Basics* – describes the basics for understanding and investigating performance questions in Adaptive Server.
 - *Locking* – describes how the various locking schemas can be used to improve performance in Adaptive Server.
 - *Optimizer and Abstract Plans* – describes how the optimizer processes queries and how abstract plans can be used to change some of the optimizer plans.
 - *Monitoring and Analyzing* – explains how statistics are obtained and used for monitoring and optimizing performance.
- *Quick Reference Guide* – provides a comprehensive listing of the names and syntax for commands, functions, system procedures, extended system procedures, datatypes, and utilities in a pocket-sized book.
- *Reference Manual* – is a series of four books that contains the following detailed Transact-SQL[®] information:
 - *Building Blocks* – Transact-SQL datatypes, functions, global variables, expressions, identifiers and wildcards, and reserved words.
 - *Commands* – Transact-SQL commands.
 - *Procedures* – Transact-SQL system procedures, catalog stored procedures, system extended stored procedures, and dbcc stored procedures.
 - *Tables* – Transact-SQL system tables and dbcc tables.

-
- *System Administration Guide* – provides in-depth information about administering servers and databases. This manual includes instructions and guidelines for managing physical resources, security, user and system databases, and specifying character conversion, international language, and sort order settings.
 - *System Tables Diagram* – illustrates system tables and their entity relationships in a poster format. Available only in print version.
 - *Transact-SQL User's Guide* – documents Transact-SQL, Sybase's enhanced version of the relational database language. This manual serves as a textbook for beginning users of the database management system. This manual also contains descriptions of the pubs2 and pubs3 sample databases.
 - *Using Adaptive Server Distributed Transaction Management Features* – explains how to configure, use, and troubleshoot Adaptive Server DTM features in distributed transaction processing environments.
 - *Using Sybase Failover in a High Availability System* – provides instructions for using the Sybase Failover software to configure an Adaptive Server as a companion server in a high availability system.
 - *Utility Guide* – documents the Adaptive Server utility programs, such as isql and bcp, which are executed at the operating system level.
 - *XA Interface Integration Guide for CICS, Encina, and TUXEDO* – provides instructions for using the Sybase DTM XA interface with X/Open XA transaction managers.
 - *XML Services in Adaptive Server Enterprise* – describes the Sybase native XML processor and the Sybase Java-based XML support, introduces XML in the database, and documents the query and mapping functions that comprise XML Services.

Other sources of information

Use the Sybase Getting Started CD, the Sybase Technical Library CD and the Technical Library Product Manuals Web site to learn more about your product:

- The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the Technical Library CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader (downloadable at no charge from the Adobe Web site, using a link provided on the CD).

- The Technical Library CD contains product manuals and is included with your software. The DynaText reader (included on the Technical Library CD) allows you to access technical information about your product in an easy-to-use format.

Refer to the *Technical Library Installation Guide* in your documentation package for instructions on installing and starting the Technical Library.

- The Technical Library Product Manuals Web site is an HTML version of the Technical Library CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Updates, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

To access the Technical Library Product Manuals Web site, go to Product Manuals at <http://www.sybase.com/support/manuals/>.

- CodeXchange is an Sybase Developer Network (SDN) feature that allows for the free exchange of code samples, utilities, scripts and other tools among SDN members. SDN CodeXchange also hosts collaborative development projects where users can work together on open source code or a tool in development.

You can access the CodeXchange on the SDN site at <http://www.sybase.com/developer/codexchange>.

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

❖ Finding the latest information on product certifications

- 1 Point your Web browser to Technical Documents at <http://www.sybase.com/support/techdocs/>.
- 2 Select Products from the navigation bar on the left.
- 3 Select a product name from the product list and click Go.
- 4 Select the Certification Report filter, specify a time frame, and click Go.
- 5 Click a Certification Report title to display the report.

❖ Creating a personalized view of the Sybase Web site (including support pages)

Set up a MySybase profile. MySybase is a free service that allows you to create a personalized view of Sybase Web pages.

- 1 Point your Web browser to Technical Documents at <http://www.sybase.com/support/techdocs/>.

- 2 Click MySybase and create a MySybase profile.

Sybase EBFs and software updates

❖ Finding the latest information on EBFs and software updates

- 1 Point your Web browser to the Sybase Support Page at <http://www.sybase.com/support>.
- 2 Select EBFs/Updates. Enter user name and password information, if prompted (for existing Web accounts) or create a new account (a free service).
- 3 Select a product.
- 4 Specify a time frame and click Go.
- 5 Click the Info icon to display the EBF/Update report, or click the product description to download the software.

Conventions

Table 1 shows the conventions for syntax statements that appear in this manual:

Table 1: Font and syntax conventions for this manual

Element	Example
Command names, procedure names, utility names, and other keywords display in sans serif font.	select sp_configure
Database names and datatypes are in sans serif font.	master database
Book names, file names, variables, and path names are in italics.	<i>System Administration Guide</i> <i>sql.ini</i> file <i>column_name</i> \$SYBASE/ASE directory
Variables—words that stand for values that you fill in—when they are part of a query or statement, are in italics in Courier font.	select <i>column_name</i> from <i>table_name</i> where <i>search_conditions</i>
Type parentheses as part of the command.	compute <i>row_aggregate</i> (<i>column_name</i>)
Double colon, equals sign indicates that the syntax is written in BNF notation. Do not type this symbol. Indicates “is defined as.”	::=
Curly braces mean that you must choose at least one of the enclosed options. Do not type the braces.	{ <i>cash, check, credit</i> }
Brackets mean that to choose one or more of the enclosed options is optional. Do not type the brackets.	[<i>cash check credit</i>]
The comma means you may choose as many of the options shown as you want. Separate your choices with commas as part of the command.	<i>cash, check, credit</i>

Element	Example
The pipe or vertical bar () means you may select only one of the options shown.	<code>cash check credit</code>
An ellipsis (...) means that you can <i>repeat</i> the last unit as many times as you like.	<pre>buy thing = price [cash check credit] [, thing = price [cash check credit]]...</pre> <p>You must buy at least one thing and give its price. You may choose a method of payment: one of the items enclosed in square brackets. You may also choose to buy additional things: as many of them as you like. For each thing you buy, give its name, its price, and (optionally) a method of payment.</p>

- Syntax statements (displaying the syntax and all options for a command) appear as follows:

```
sp_dropdevice [device_name]
```

For a command with more options:

```
select column_name
from table_name
where search_conditions
```

In syntax statements, keywords (commands) are in normal font and identifiers are in lowercase. Italic font shows user-supplied words.

- Examples showing the use of Transact-SQL commands are printed like this:

```
select * from publishers
```

- Examples of output from the computer appear as follows:

```
pub_id      pub_name                city                state
-----
0736       New Age Books           Boston              MA
0877       Binnet & Hardley        Washington           DC
1389       Algodata Infosystems   Berkeley            CA
```

(3 rows affected)

In this manual, most of the examples are in lowercase. However, you can disregard case when typing Transact-SQL keywords. For example, `SELECT`, `Select`, and `select` are the same.

Adaptive Server's sensitivity to the case of database objects, such as table names, depends on the sort order installed on Adaptive Server. You can change case sensitivity for single-byte character sets by reconfiguring the Adaptive Server sort order. For more information, see the *System Administration Guide* for Adaptive Server.

If you need help

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online help, have the designated person contact Sybase Technical Support or the Sybase subsidiary in your area.

PART 1

Introduction to Web Services

This part introduces Adaptive Server Enterprise Web Services.

Understanding Adaptive Server Enterprise Web Services

This chapter discusses the following:

Topic	Page
Overview	3
Components of Adaptive Server Enterprise Web Services	4
Advantages of Adaptive Server Enterprise Web Services	4
Web services standards	6

Overview

A Web service is a self-contained, modular application that can be accessed through a network connection. Using a Web Service, the end user trades performance for increased interoperability enforced by adherence to the Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), HTML, and Extensible Markup Language (XML) open standards.

Regardless of the programming language in which it has been implemented, a Web service can be accessed from many different platforms and operating systems, thus greatly enhancing the ability for diverse applications to share data. By using many discrete Web services, each handling a limited set of specific tasks, business enterprises can dynamically and incrementally integrate by exposing their existing software in a secure and controlled environment. By providing a standardized means to invoke remote applications, Web services reduce the amount of code required for infrastructure. By enabling users to extract implementation from exposed interfaces (WSDL), Web services provide the tools needed to build a service-oriented architecture (SOA).

Components of Adaptive Server Enterprise Web Services

Adaptive Server Enterprise Web Services consists of two components, a Web Services Producer and a Web Services Consumer. Both components run independently of Adaptive Server Enterprise and are enabled by the same license.

- The Web Services Producer component enables client applications to access SQL and stored procedures in Adaptive Server Enterprise using SOAP. For more information on the Web Services Producer, see Chapter 2, “Understanding the Web Services Producer.”
- The Web Services Consumer component enables Adaptive Server Enterprise to access the Web services of other applications. These external Web services are mapped to an Adaptive Server Enterprise proxy table at runtime. For more information on the Web Services Consumer, see Chapter 3, “Understanding the Web Services Consumer.”

Advantages of Adaptive Server Enterprise Web Services

With the Web Services Producer, the user can use stored procedures, user-defined functions, and SQL to query and manipulate data. A client application can send a SOAP request containing SQL commands and receive results through SOAP. Data is returned according to the SQLX standard, and the client application can receive XML data, schema, and DTDs.

Adaptive Server Enterprise Web Services provides tools that enable the user to predict the cost of integrating new data, reconcile differing data sources into a common data model, and reduce the cost of developing, deploying, and maintaining applications.

Stored procedures and functions

Stored procedures separate the internal, logical view of the data from business-level logic and extend the influence and performance of SQL. Stored procedures can also be executed remotely. The user can use both stored procedures and user-defined functions to invoke Java methods, as specified in the ANSI SQLJ standard, and to retrieve data in standard XML format.

SQL

Because SQL can be used to manipulate XML data, SOAP-enabled client applications can use the Web Services Producer to manage data in Adaptive Server Enterprise. SQL can also be used to invoke Web services through the Web Services Consumer.

Security

The Web Services security features include Secure Sockets Layer (SSL) and provide important database security and authorization features, like access control through the Lightweight Directory Access Protocol (LDAP).

LDAP

LDAP is an Internet protocol for accessing directories in a distributed environment. An LDAP server stores the user information needed to establish connections between resources and grant access to directories, eliminating the need for client applications to know this information. Adaptive Server Enterprise Web Services enables client applications to access Web methods using LDAP.

Adaptive Server Enterprise Web Services supports LDAP version 3 servers. For more detailed information on using LDAP to enable user authentication and to locate Adaptive Server Enterprise data servers, see the *System Administration Guide* for Adaptive Server Enterprise.

Web services standards

Web services are structured with XML, described with WSDL, and transferred with SOAP over HTTP. Adaptive Server Enterprise Web Services enables client applications to access Web services and can consume remote Web services.

XML

XML is used to describe data. XML is derived from SGML and possesses some qualities of other markup languages, like HTML. However, XML is extensible because its tags are user-defined, making it ideal for exchanging data in a structure that is intelligible to two or more communicating applications.

Example

The following isql query to the pubs2 database finds information on discounts:

```
1> select * from discounts
2> go
```

This query produces the following result set:

discounttype	stor_id	lowqty	highqty	discount
Initial Customer	NULL	NULL	NULL	10.500000
Volume Discount	NULL	100	1000	6.700000
Huge Volume Discount	NULL	1001	NULL	10.000000
Customer Discount	8042	NULL	NULL	5.000000

This result set can be represented in XML in many ways. The following is an XML representation produced by Adaptive Server Enterprise Web Services and formatted in SQLX, which is part of the ANSI standard for SQL:

```
<?xml version="1.0" encoding="UTF-8">
<ws xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <row>
    <discounttype>Initial Customer</discounttype>
    <discount>10.5</discount>
  </row>
  <row>
    <discounttype>Volume Discount</discounttype>
```

```
        <lowqty>100</lowqty>
        <highqty>1000</highqty>
        <discount>6.7</discount>
    </row>
    <row>
        <discounttype>Huge Volume Discount
    </discounttype>
        <lowqty>1001</lowqty>
        <discount>10.0</discount>
    </row>
    <row>
        <discounttype>Customer Discount</discounttype>
        <stor_id>8042</stor_id>
        <discount>5.0</discount>
    </row>
</ws>
```

The initial line describes the XML version and character encoding. The remaining tags are user-defined and describe both the structure and data of the document. These user-defined tags enable documents to be customized for a specific application, such as one that uses discount information to compute prices.

XML document structure

The user-defined elements and their arrangement in a well-formed XML document is defined either by a Document Type Definition (DTD) or an XML schema.

Following is a DTD for the previous example for discount information:

```
<!DOCTYPE ws [
  <!ELEMENT ws (row*)>
  <!ELEMENT row (discounttype, stor_id?, lowqty?,
highqty?, discount)>
  <!ELEMENT discounttype (#PCDATA)>
  <!ELEMENT stor_id (#PCDATA)>
  <!ELEMENT lowqty (#PCDATA)>
  <!ELEMENT highqty (#PCDATA)>
  <!ELEMENT discount (#PCDATA)>
]>
```

The following is part of an XML schema for the previous example for discount information:

```
<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
xmlns:sqlxml="http://www.iso-
standards.org/mra/9075/sqlx">
<xsd:import
  namespace="http://www.w3.org/2001/XMLSchema"
  schemaLocation="http://www.iso-
standards.org/mra/9075/sqlx.xsd" />
<xsd:complexType name="RowType.ws">
  <xsd:sequence>
    <xsd:element name="discounttype"
      type="VARCHAR_40" />
    <xsd:element name="stor_id" type="CHAR_4"
      minOccurs="0" maxOccurs="1"/>
    <xsd:element name="lowqty" type="SMALLINT"
      minOccurs="0" maxOccurs="1"/>
    <xsd:element name="highqty" type="SMALLINT"
      minOccurs="0" maxOccurs="1"/>
    <xsd:element name="discount" type="DOUBLE" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="TableType.ws">
  <xsd:sequence>
    <xsd:element name="row" type="RowType.ws"
      minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="VARCHAR_40">
  <xsd:restriction base="xsd:string">
    <xsd:length value="40"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="SMALLINT">
  <xsd:restriction base="xsd:integer">
    <xsd:maxInclusive value="32767"/>
    <xsd:minInclusive value="-32768"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="DOUBLE">
  <xsd:restriction base="xsd:double"/>
</xsd:simpleType>
  <xsd:element name="ws" type="TableType.ws"/>
</xsd:schema>
```

An XML schema or DTD can be included as part of the XML document they describe or referenced as separate files. The respective file suffixes for an XML schema and a DTD are *.xsd* and *.dtd*.

For more detailed information on XML, refer to the following documents:

- World Wide Web Consortium (W3C), at <http://www.w3.org>
- W3C, Extensible Markup Language (XML), at <http://www.w3.org/XML/>

WSDL

A WSDL document is written in XML and describes a Web service. In addition to specifying the location of the Web service, a WSDL description also specifies the methods provided by the Web service, and the messages, datatypes, and communication protocols used by the Web service with the following tags:

- `<service>` – defines the name of the Web service. For example, a Web service called `ExecuteStoredProcService` could be named as follows:

```
<wsdl:service name="ExecuteStoredProcService">
  <wsdl:port binding="impl:aseSoapBinding" name="ase">
    <wsdlsoap:address location="http://myserver:8181/services/ase"/>
  </wsdl:port>
</wsdl:service>
```

A WSDL document may contain one or more `<service>` tags. In the case of the Web Services Producer, there is only one service, which is named “ase.”

- `<binding>` – defines the communication protocols used. The following example uses the SOAP protocol:

```
<wsdl:binding name="aseSoapBinding" type="impl:ExecuteStoredProc">
...
</wsdl:binding>
```

WSDL also supports use of HTTP and MIME protocols.

- `<port>` – specifies the Web service address. For example:

```
<wsdl:port binding="impl:aseSoapBinding" name="ase">
  <wsdlsoap:address location="http://myserver:8181/services/ase"/>
</wsdl:port>
```

The `<port>` tag has attributes for name and binding.

- `<message>` – defines the messages used. For example:

```
<wsdl:message name="executeRequest">
  <wsdl:part name="service" type="xsd:string"/>
  <wsdl:part name="userName" type="xsd:string"/>
  <wsdl:part name="password" type="xsd:string"/>
```

```
<wsdl:part name="sqlOptions" type="xsd:string"/>
<wsdl:part name="sql" type="xsd:string"/>
</wsdl:message>
```

This is a request message for a method called `executeRequest`. The `<part>` tags correspond to parameter values for the method call in a request message and to return values in a response.

- `<operation>` – associates a message with a Web method request or response. For example:

```
<wsdl:operation name="execute" parameterOrder="service userName
password sqlOptions sql">
  <wsdl:input message="impl:executeRequest" name="executeRequest"/>
  <wsdl:output message="impl:executeResponse" name="executeResponse"/>
</wsdl:operation>
```

- `<portType>` – defines the methods provided. The `<operation>` tag is a child element of `<portType>`. For example:

```
<wsdl:portType name="ExecuteStoredProc">
  <wsdl:operation name="execute" parameterOrder="aseServerName
asePortNumber
. . . .
  </wsdl:operation>
</wsdl:portType>
```

- `<types>` – defines the datatypes used. WSDL uses XML schema syntax to define datatypes.

WSDL is usually automatically generated by the Web Services Producer component and can be viewed in a Web browser at the following location:

`http://myserver:producer_port/services/ase?wsdl`

where *myserver* is the name of the host on which the Web Services Producer component is running, and *producer_port* is the port number.

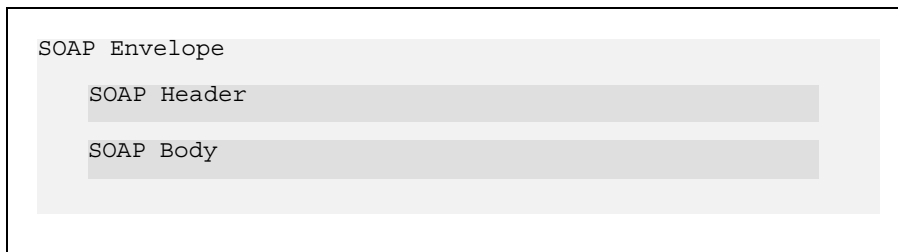
SOAP

SOAP is a platform- and language-independent protocol based on XML and used to send messages and data between applications. SOAP defines the structure of messages, describes how messages are to be processed, and provides rules for encoding application-defined datatypes. SOAP allows applications to send and receive remote procedure calls (RPCs) using any standard transport-layer protocol, usually HTTP.

SOAP message structure

A SOAP message consists of a header and a body, both of which are contained in a SOAP envelope:

Figure 1-1: SOAP message structure



Following is an example of SOAP request message for a select statement.

Figure 1-2: SOAP request message

```

POST /services/ase HTTP/1.0
Content-Type: text/xml; charset=utf-8
Accept: application/soap+xml, application/dime, multipart/related, text/*
User-Agent: Axis/1.1RC2
Host: localhost
Cache-Control: no-cache
Pragma: no-cache
SOAPAction: ""
Content-Length: 674

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:execute
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="urn:genwsdl.ws.ase.sybase.com">
      <service xsi:type="xsd:string">myservice</service>
      <userName xsi:type="xsd:string">sa</userName>
      <password xsi:type="xsd:string"></password>
      <sqlxOptions xsi:type="xsd:string">tablename=ws</sqlxOptions>
      <sql xsi:type="xsd:string">select @@version</sql>
    </ns1:execute>
  </soapenv:Body>
</soapenv:Envelope>

```

The request message contains no header information. The following SOAP response message corresponds to the previous request message. This message contains a header. The body of the message is not shown.

Figure 1-3: SOAP response message

```
HTTP/1.1 200 OK
Date: Wed, 1 Oct 2003 22:02:10 GMT
Server: Jetty/4.2.1 (Windows 2000 5.0 x86)
Content-Type: text/xml; charset=utf-8

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <ns1:sessionID soapenv:actor="" soapenv:mustUnderstand="0"
      xsi:type="xsd:long"
      xmlns:ns1="http://xml.apache.org/axis/session">-6707797562660015345
    </ns1:sessionID>
  </soapenv:Header>
  <soapenv:Body>
    ....
  </soapenv:Body>
</soapenv:Envelope>
```

For more detailed information on SOAP, refer to the following documents:

- Simple Object Access Protocol (1.1) <http://www.w3.org/TR/SOAP/>
- Simple Object Access Protocol (1.2) Part 1:
<http://www.w3.org/TR/soap12-part1/>
- Simple Object Access Protocol (1.2) Part 2:
<http://www.w3.org/TR/soap12-part2/>

Understanding the Web Services Producer

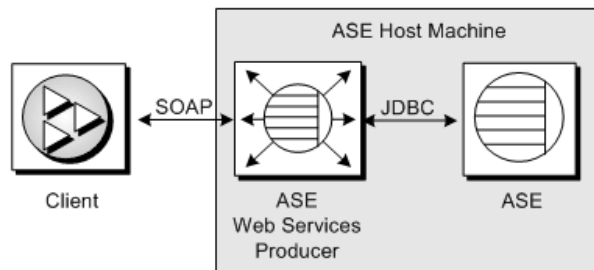
This chapter discusses the following:

Topic	Page
Overview	15
Components	16

Overview

The Web Services Producer enables a client application to access Adaptive Server Enterprise stored procedures and SQL using SOAP. The Web Services Producer component runs on the same machine as Adaptive Server Enterprise. The output of the Web Services Producer component complies with SQLX, which is defined as part of the ANSI specification for SQL.

Figure 2-1: Web Services Producer

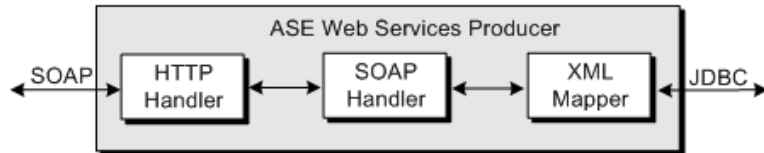


The client can send a SQL or stored procedure command as a SOAP request, and any result is returned as a SOAP response. The data in the SOAP response conforms to the SQLX standard.

Components

The Web Services Producer consists of three components: an HTTP handler, a SOAP handler, and an XML mapper.

Figure 2-2: Web Services Producer components



HTTP handler

The HTTP handler supports HTTP 1.1 and listens for SOAP requests sent using the HTTP POST and GET methods. The HTTP handler also supports SSL connections.

The HTTP handler is based on the Jetty Web server. For information on the Jetty Web server, refer to the documentation provided at <http://jetty.mortbay.org>.

SOAP handler

The SOAP handler supports SOAP 1.2 and processes SOAP requests. The SOAP handler also generates WSDL files describing Web services.

The SOAP handler is based on the Apache Axis SOAP engine. For information on Apache Axis, see the documentation provided at <http://ws.apache.org/axis/>.

XML mapper

The XML mapper encodes relational data, returned from Adaptive Server Enterprise through JDBC, into XML that complies with the SQLX standard. The XML mapper also generates a DTD and an XML schema to describe the data.

Understanding the Web Services Consumer

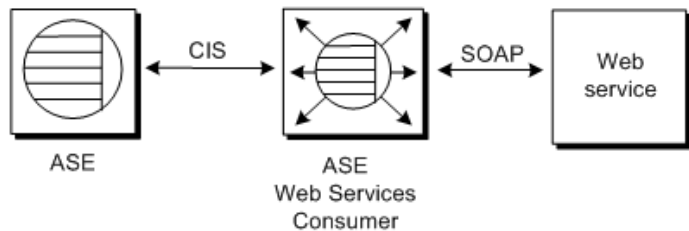
This chapter discusses the following:

Topic	Page
Overview	17
Components	18

Overview

The Web Services Consumer enables Adaptive Server Enterprise to access and execute Web methods. A Web method is made accessible by mapping it to an Adaptive Server Enterprise proxy table using information provided in the WSDL file for the Web method. A Web method can then be invoked with a select on the proxy table.

Figure 3-1: Web Services Consumer

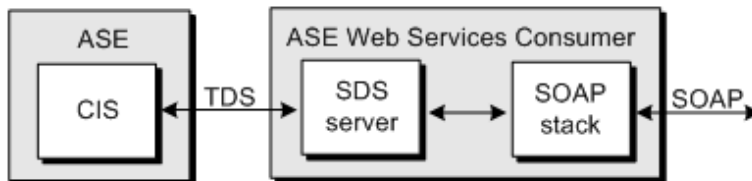


Note The Web service accessed may reside within or beyond a firewall.

Components

The Web Services Consumer consists of a Specialty Data Store (SDS) server and a WSDL parser. The SDS is used as a Component Integration Service (CIS) to map the Web method to a proxy table. The proxy table is constructed using a WSDL file parsed with Apache Axis.

Figure 3-2: Web Services Consumer components



SDS server

When Adaptive Server Enterprise receives a select statement for a Web method proxy table, Adaptive Server Enterprise forwards the request to the Web Services Consumer in a Tabular Data Stream (TDS). The SDS server, which acts as a server for CIS, allows the Web Services Consumer to intercept and handle the TDS from Adaptive Server Enterprise.

SOAP stack

The SOAP stack is a layered set of functionalities that collectively handle the serialization and transport of XML-encoded data. The SOAP stack uses the WSDL file for a Web method to determine the structure of a corresponding proxy table. The SOAP stack also generates SOAP requests corresponding to the select statement submitted to Adaptive Server Enterprise and sends these SOAP requests to a SOAP server. The SOAP stack supports both RPC/encoded and document/literal Web methods.

RPC/encoded

The SOAP messages for an RPC/encoded Web method contain an XML element for each method parameter. Messages for an RPC/encoded Web method are encoded according to the SOAP specification. The proxy table representing an RPC/encoded Web method contains a column for each input and output parameter.

document/literal

In the SOAP messages for a document/literal Web method, the communicating parties specify the data being transmitted and formatted according to XML schemas incorporated into the WSDL file. Messages for a document/literal Web method are serialized and deserialized according to the WSDL file for the Web method. The proxy table representing a document/literal Web method contains two columns, `_inxml` and `outxml`.

The Web Services Interoperability (WSI) Organization, which defines practical interoperability guidelines, recommends using document/literal Web methods to enhance portability. Sybase supports this recommendation.

PART 2

Configuration

This part provides instructions for configuring Web Services in Adaptive Server.

Configuring Adaptive Server Enterprise Web Services

Adaptive Server Enterprise Web Services is installed as part of the installation for Adaptive Server Enterprise. Adaptive Server Enterprise Web Services may be configured during the installation of Adaptive Server Enterprise using a configuration wizard or after the installation from Sybase Central. For instructions on installing Adaptive Server Enterprise, see the *Installation Guide* for Adaptive Server Enterprise.

Note Unless otherwise specified, directories listed in this and subsequent chapters are assumed to reside under the `$$SYBASE/WS-12_5` directory on UNIX and the `%SYBASE%\WS-12_5` directory on Windows.

This chapter covers the following topics:

Topic	Page
Configuration	23
Configuration files	26
Security	26

Configuration

Adaptive Server Enterprise Web Services is configured with the Configuration Utility. You can use the Configuration Utility either during or after the installation of Adaptive Server Enterprise Web Services:

- Configuring during installation – You have the option of configuring Adaptive Server Enterprise Web Services during a custom installation of the Adaptive Server Enterprise Web Services feature.

- Configuring after installation – If you do not configure the Web Services feature during installation, you must configure the feature from Sybase Central after the installation is complete. Sybase Central provides a GUI Configuration Utility similar to that used in the installation procedure. You can also configure the Web Services feature in silent mode or in command/console mode.

Configuring during installation

To configure Web Services, activate the Configuration Utility during the installation of the Web Services feature, and follow the steps in the wizard.

Note After your installation and before using Sybase Central, make sure that you have entered license keys for the Web Services feature. For information on licensing, see “Licensing” on page 25.

Configuring after installation

If you have already installed Adaptive Server Enterprise Web Services, you can start the Configuration Utility from the Adaptive Server Enterprise plug-in to Sybase Central. You can configure Web Services using the installation GUI, or you can use the command/console or silent modes.

Note Before using Sybase Central, make sure that you have entered license keys for the Web Services feature. For information on licensing, see “Licensing” on page 25.

❖ Starting the Configuration Utility GUI from Sybase Central

- 1 Click on the Utilities folder in the Folders view for the Adaptive Server Enterprise plug-in of Sybase Central.
- 2 Find the Configure Web Service icon in the details view to the right of the Folders view. Select the icon for Configure Web Service to start the Configuration Utility.
- 3 Follow the steps in the wizard.

❖ Starting the Configuration Utility in command/console mode

- 1 Open a console window.
- 2 Enter the following command:

```
aseplugin -I
```
- 3 Follow the steps indicated on the console.

❖ Starting the Configuration Utility in silent mode

- 1 Edit the entries in your *myres.properties* file to indicate the desired values. To set a property, add “=” and the property value to the *myres.properties* entry. For details on the contents of the *myres.properties* file, see “myres.properties” in the Appendix, “Configuration Properties.”
- 2 Open a console window.
- 3 Enter the following command:

```
aseplugin -s path
```

where *path* is the path to your *myres.properties* file.

In silent mode, no further user action is necessary to configure Web Services.

Licensing

Although installed with Adaptive Server Enterprise, Adaptive Server Enterprise Web Services is a separately licensed feature requiring a license registered through SySAM named ASE_WEBSERVICES. Before configuring Adaptive Server Enterprise Web Services, do the following:

- 1 Make sure that your license is registered through SySAM. For more information on SySAM, see the *Installation Guide* for Adaptive Server Enterprise.
- 2 Establish an isql session with your Adaptive Server Enterprise.
- 3 Activate your license by entering the following command in isql:

```
sp_configure 'enable webservices', 1
```

Configuration files

The *props* directory contains the following configuration files:

- *ws.properties*
- *logging.properties*
- *wsmmsg.properties*

ws.properties

The *ws.properties* file contains configuration settings for Adaptive Server Enterprise Web Services. For details on these configuration properties, see Appendix B, “Configuration Properties.”

logging.properties

The *logging.properties* file defines where logging output is sent. You can direct output to a log file or to the console. If output is directed to a log file, you can specify when to roll over to a new log file. You can also control the format of logging messages. For information on log rollover policies, see “Rolling over log files” on page 55.

The *logging.properties* file entries and logging behavior follow that for the Apache log4j package. For detailed information, refer to the documentation for log4j at <http://jakarta.apache.org/log4j/docs/documentation.html>.

wsmmsg.properties

The *wsmmsg.properties* file is for internal use by Adaptive Server Enterprise Web Services.

Security

To ensure secure operation of Adaptive Server Enterprise Web Services, Sybase recommends that you do the following:

- Install the Web Services Producer component of Adaptive Server Enterprise Web Services on the same machine as Adaptive Server Enterprise.
- Use LDAP for user authentication in Adaptive Server Enterprise.
- Use SSL to connect to the Web Services Producer component of Adaptive Server Enterprise Web Services. For instructions on configuring SSL, see “Configuring SSL.”

Configuring SSL

Note Two certificate passwords are created by default as “sybase” from InstallShield during installation. You can change these from Sybase Central later.

SSL is configured automatically using the Configuration Utility from InstallShield or from Sybase Central. However, you can also configure SSL manually. To manually configure SSL for Adaptive Server Enterprise Web Services, run the *configssl* script, which can be found in the *bin* directory:

```
configssl -d <domain_hostName> -k <keystore> -h  
<httpsPort> -f <property_file> -c  
<certificate_password> -s <keystore_password>
```

where:

- *domain_hostName* is the host name of the URL to connect to using SSL. For example, the *domain_hostName* for the following URL would be *mydomainhostname*:

```
http://mydomainhostname:8181/services/ase
```

There is no default for this parameter value.

- *keystore* is the location and file at which to store certificates. The default location for UNIX is *\$\$SYBASE/WS-12_5/props/keystore*, or *%SYBASE%\WS-12_5\props\keystore* for Windows.
- *httpsPort* is the port on which to listen for an SSL connection. The default is 8182.
- *property_file* is the location and name of the properties file to update. The default location for UNIX is *\$\$SYBASE/WS-12_5/props/ws.properties*, or *%SYBASE%\WS-12_5\props\ws.properties* for Windows.

- *certificate_password* is the password for the certificate. There is no default for this parameter value. If no password is supplied when the script is invoked, the script will prompt for a value.
- *keystore_password* is the password for the keystore. There is no default for this parameter value. If no password is supplied when the script is invoked, the script will prompt for a value.

Note You can also add your own certificate for SSL. For instructions on how to add your own certificate, see the documentation for the *keytool* utility in the JRE that manipulates the *keystore* file. The JRE delivered with Adaptive Server Enterprise Web Services is version 1.3.

Installing a certificate for Microsoft .NET

A Microsoft .NET client requires a certificate to access the Producer component using SSL. Use the following procedure to install a certificate for Microsoft .NET.

❖ Installing a certificate for Microsoft .NET

- 1 Start the Web Services Producer component with SSL. For instructions on starting the Web Services Producer component, see “Starting the Web Services Producer” on page 33.

- 2 Enter the following in the Address bar of Microsoft Internet Explorer:

`https://<producer_host>:<SSL_port>`

where *producer_host* is the host on which the Web Services Producer component runs, and *SSL_port* is the port for the Web Services Producer component.

The Security Alert dialog box appears.

- 3 Click View Certificate. The Certificate dialog box appears.
- 4 Click Install Certificate. The Certificate Manager Import wizard opens.
- 5 Click Next until the Certificate Manager Import wizard indicates that the certificate was successfully installed and returns to the Certificate dialog box.
- 6 Click OK. The browser returns you to the Security Alert dialog box.

- 7 Click Yes. The browser window should display a page titled “Welcome to the ASE Web Services Producer.”

❖ **Verifying the certificate installation**

- 1 Close all browser windows.
- 2 Restart Microsoft Internet Explorer.
- 3 Enter the following in the Address bar of Microsoft Internet Explorer:

`https://<producer_host>:<SSL_port>`

where *producer_host* is the host on which the Web Services Producer component runs, and *SSL_port* is the port for the Web Services Producer component.

No Security Alert dialog box should appear.

PART 3

Administration

This part describes product functionality and administrative tasks.

Using Adaptive Server Enterprise Web Services

This chapter covers the following topics:

Topic	Page
Using the Web Services Producer	33
Using the Web Services Consumer	42
Adaptive Server Enterprise Web Services Logging	54
Sybase Central	55
Using the Web interface	56

Before using Adaptive Server Enterprise Web Services, make sure you have completed the configuration tasks in Chapter 4, “Configuring Adaptive Server Enterprise Web Services.”

Using the Web Services Producer

This section documents the following:

- Starting the Web Services Producer
- Stopping the Web Services Producer
- Adaptive Server Enterprise Web Services methods

Starting the Web Services Producer

To start the Web Services Producer component for Adaptive Server Enterprise Web Services, execute the *runproducer* script, which is located in the *bin* directory:

```
runproducer -U <ase_username> -P <ase_password> -S
<ase_service_name> -f <property_file> -v
```

where:

- *ase_username* is the user name for the Adaptive Server Enterprise. There is no default for this parameter value. If you do not supply a value for this parameter, you will be prompted for one.
- *ase_password* is the password for the Adaptive Server Enterprise. There is no default for this parameter value. If you do not supply a value for this parameter, you will be prompted for one.
- *ase_service_name* is the name of the Web service. There is no default for this parameter value. If you do not supply a value for this parameter, you will be prompted for one.
- *property_file* is the location and name of the properties file to update. The default location for UNIX is *SYBASE/WS-12_5/props/ws.properties*, or *%SYBASE%\WS-12_5\props\ws.properties* for Windows.
- *-v* specifies that the Producer component should display version information on start-up.

You can start both the Producer and Consumer components for Adaptive Server Enterprise Web Services by executing the *runws* script, which is located in the *bin* directory:

```
runws -U <ase_username> -P <ase_password> -S  
<ase_service_name> -f <property_file> -v
```

where the parameter values are the same as for the *runproducer* script.

Conditions

The Web Services Producer component will start if the following conditions are met:

- The *ase_service_name* provided is located on an LDAP server pointed to by the *libtcl.cfg* file or in the *interfaces* file for Adaptive Server Enterprise.

Adaptive Server Enterprise Web Services first searches for an entry containing the value of *ase_service_name* on an LDAP server pointed to by the *libtcl.cfg* file. Adaptive Server Enterprise Web Services locates the *libtcl.cfg* file using the `com.sybase.ase.ws.libtcl` entry in the *ws.properties* file. If no entry is found on an LDAP server, Adaptive Server Enterprise Web Services looks for an entry in the *interfaces* file for Adaptive Server Enterprise.

Note On Windows systems, the *interfaces* file is named *sql.ini*.

Adaptive Server Enterprise Web Services locates the *interfaces* file using the `com.sybase.ase.ws.interfaces` entry in the *ws.properties* file.

- A successful login can be made using the *ase_username* and *ase_password* provided.

Note The password for an Adaptive Server Enterprise user may be set to a null string. For null passwords on UNIX, use the token `nopasswordspecified` anywhere the password is required. On Windows, you can use the empty string, `""`. Sybase does not recommend using null passwords for security.

- The login to Adaptive Server Enterprise has sa role privileges.
- The following stored procedure command has been executed in isql for your Adaptive Server Enterprise:

```
sp_configure 'enable webservices', 1
```

Verification

After successfully executing the *runproducer* script, verify that Adaptive Server Enterprise Web Services is enabled and that the Web Services Producer component is running.

❖ **Verifying that Adaptive Server Enterprise Web Services is enabled**

To verify that Adaptive Server Enterprise Web Services is enabled:

- Execute the following command on Adaptive Server Enterprise:

```
sp_configure 'enable webservices'
```

If `sp_configure` returns a value of 1, the Web Services feature has been enabled. A return value of 0 indicates the feature is not enabled.

❖ **Verifying that the Web Services Producer component is running**

- Check the *producer.log* file in the *logs* directory for messages indicating that the Web Services Producer component is running. For example:

```
2004-03-29 16:29:29.522 INFO [main] - Starting HTTP Server on Port: 8181
```

For SSL, the log indicates an HTTPS port and related SSL information. For example:

```
2004-03-29 16:29:29.532 INFO [main] - Https Port [8182], KeyPassword: ...
```

Stopping the Web Services Producer

To stop the Web Services Producer component for Adaptive Server Enterprise Web Services, execute the *stopproducer* script, which is located in the *bin* directory:

```
stopproducer -U <ase_username> -P <ase_password> -S  
<ase_service_name> -f <property_file> -v
```

where:

- *ase_username* is the user name for the Adaptive Server Enterprise. There is no default for this parameter value. If you do not supply a value for this parameter, you will be prompted for one.
- *ase_password* is the password for the Adaptive Server Enterprise. There is no default for this parameter value. If you do not supply a value for this parameter, you will be prompted for one.
- *ase_service_name* is the name of the Web service. There is no default for this parameter value. If you do not supply a value for this parameter, you will be prompted for one.
- *property_file* is the location and name of the properties file to update. The default location for UNIX is `$(SYBASE)/WS-12_5/props/ws.properties`, or `%SYBASE%\WS-12_5\props\ws.properties` for Windows.
- `-v` specifies that the Producer component should display version information at shut-down.

You can stop both the Producer and Consumer components for Adaptive Server Enterprise Web Services by executing the *stopws* script, which is located in the *bin* directory:

```
stopws -U <ase_username> -P <ase_password> -S  
<ase_service_name> -f <property_file> -v
```

where the parameter values are the same as for the *stopproducer* script.

Conditions

The Web Services Producer component will stop if the following conditions are met:

- The *ase_service_name* provided is located on an LDAP server pointed to by the *libtcl.cfg* file or in the *interfaces* file for Adaptive Server Enterprise.

Adaptive Server Enterprise Web Services first searches for an entry containing the value of *ase_service_name* on an LDAP server pointed to by the *libtcl.cfg* file. Adaptive Server Enterprise Web Services locates the *libtcl.cfg* file using the `com.sybase.ase.ws.libtcl` entry in the *ws.properties* file. If no entry is found on an LDAP server, Adaptive Server Enterprise Web Services looks for an entry in the *interfaces* file for Adaptive Server Enterprise.

Note On Windows systems, the *interfaces* file is named *sql.ini*.

Adaptive Server Enterprise Web Services locates the *interfaces* file using the `com.sybase.ase.ws.interfaces` entry in the *ws.properties* file.

- A successful login can be made using the *ase_username* and *ase_password* provided.

Note The password for an Adaptive Server Enterprise user may be set to a null string. For null passwords on UNIX, use the token `nopasswordspecified` anywhere the password is required. On Windows, you can use the empty string, `""`. Sybase does not recommend using null passwords for security.

- The login to Adaptive Server Enterprise has *sa* role privileges.

Adaptive Server Enterprise Web Services methods

To access Adaptive Server Enterprise Web Services, your client must use the methods exposed by the Web Services Producer component. These methods are mapped in SOAP as rpc:

```
<soap:binding style="rpc" ...>
```

Message data is encoded:

```
<soap:body use="encoded" ...>
```

The Web Services Producer component provides the following methods:

- `execute` – executes a SQL statement or stored procedure.
- `login` – establishes a persistent connection to Adaptive Server Enterprise.
- `logout` – explicitly terminates an Adaptive Server Enterprise connection.

The syntax for these methods is the same regardless of whether they are invoked using HTTP or SSL.

execute

The `execute` method executes a Transact-SQL statement or stored procedure in Adaptive Server Enterprise.

Syntax

```
execute aseServerName userName password sqlOptions  
sql
```

Parameters

- *aseServerName*
SOAP string indicating the name of the Adaptive Server Enterprise server in the *interfaces* file or LDAP server.

At each invocation of the `execute` method, Adaptive Server Enterprise Web Services uses the value of *aseServerName* in the following way:

- a Adaptive Server Enterprise Web Services searches for an entry containing the value of *aseServerName* on an LDAP server pointed to by the *libtcl.cfg* file. Adaptive Server Enterprise Web Services locates the *libtcl.cfg* file using the `com.sybase.ase.ws.libtcl` entry in the *ws.properties* file.

- b If no entry is found on an LDAP server, Adaptive Server Enterprise Web Services looks for an entry in the *interfaces* file for Adaptive Server Enterprise.

Note On Windows systems, this file is named *sql.ini*.

Adaptive Server Enterprise Web Services locates the *interfaces* file using the `com.sybase.ase.ws.interfaces` entry in the *ws.properties* file.

- c If no entry is found in the *interfaces* file for Adaptive Server Enterprise, execution fails for the `execute` method.
- *userName*
SOAP string indicating the user ID needed to log in to the Adaptive Server Enterprise.
 - *password*
SOAP string indicating the password needed to log in to the Adaptive Server Enterprise.
 - *sqlxOptions*
SOAP string indicating one or more option parameters. These parameters specify characteristics of the SQLX result set. The following are valid option parameters:
 - `binary={hex | base64}`
 - `columnstyle={element | attribute}`
 - `format={yes | no}`
 - `header={yes | no}`
 - `nullstyle={attribute | omit}`
 - `prefix="value"`
 - `root={yes | no}`
 - `rowname="value"`
 - `schemaloc="value"`
 - `statement={yes | no}`
 - `tablename="value"`
 - `targetns="value"`

You must provide a value for *value*. For more information on SQLX functions and options, see *XML Services in Adaptive Server Enterprise*.

- *sql*

SOAP string indicating the SQL statement or stored procedure to be executed on Adaptive Server Enterprise.

Example 1

This example checks the version number for Adaptive Server Enterprise.

```
execute johndoe-sun sa password "tablename=ws" "select @@version"
```

This example invokes the Web method directly. Adaptive Server Enterprise Web Services returns an XML schema, a DTD, and a result set containing the result of the executed statement.

Example 2

This example computes a left join on tables in the pubs2 database.

```
execute johndoe-sun sa password "tablename=ws"
"select title, price, au_fname, au_lname from (titles
left join titleauthor on titles.title_id =
titleauthor.title_id ) left join authors on
titleauthor.au_id = authors.au_id and titles.price >
$15.00"
```

login

The login method establishes a persistent connection to Adaptive Server Enterprise.

Syntax

```
login aseServerName userName password
```

Parameters

- *aseServerName*

SOAP string indicating the name of the Adaptive Server Enterprise on which to execute the SQL statement or stored procedure.

At each invocation of the execute method, Adaptive Server Enterprise Web Services uses the value of *aseServerName* in the following way:

- a Adaptive Server Enterprise Web Services searches for an entry containing the value of *aseServerName* on an LDAP server pointed to by the *libtcl.cfg* file.

- b If no entry is found on an LDAP server, Adaptive Server Enterprise Web Services looks for an entry in the *interfaces* file for Adaptive Server Enterprise.

Note On Windows systems, this file is named *sql.ini*.

Adaptive Server Enterprise Web Services locates the *interfaces* file using the `com.sybase.ase.ws.interfaces` entry in the *ws.properties* file.

- c If no entry is found in the *interfaces* file for Adaptive Server Enterprise, execution fails for the `execute` method.

- *username*

SOAP string indicating the user ID needed to log in to the Adaptive Server Enterprise.

- *password*

SOAP string indicating the password needed to log in to the Adaptive Server Enterprise.

Usage

Before a SQL statement or stored procedure can be executed on Adaptive Server Enterprise, a connection must first be established. However, the `login` method is optional. If you invoke an `execute` method without first invoking the `login` method, Adaptive Server Enterprise Web Services automatically establishes a non-persistent connection to Adaptive Server Enterprise. The `login` method initiates a persistent connection to Adaptive Server Enterprise. The connection is terminated with the `logout` method. Persistent connections that are inactive for 60 seconds are terminated automatically.

logout

The `logout` method terminates a persistent connection to Adaptive Server Enterprise.

Syntax

`logout`

Usage

The `logout` method terminates a persistent connection to Adaptive Server Enterprise established by the `login` method.

Using the Web Services Consumer

This section documents the following:

- Starting the Web Services Consumer
- Stopping the Web Services Consumer
- Using `sp_webservices`
- Invoking a Web service

Starting the Web Services Consumer

To start the Web Services Consumer component for Adaptive Server Enterprise Web Services, execute the `runconsumer` script, which is located in the `bin` directory:

```
runconsumer -U <ase_username> -P <ase_password> -S  
<ase_service_name> -f <property_file> -v
```

where:

- `ase_username` is the user name for the Adaptive Server Enterprise. There is no default for this parameter value. If you do not supply a value for this parameter, you will be prompted for one.
- `ase_password` is the password for the Adaptive Server Enterprise. There is no default for this parameter value. If you do not supply a value for this parameter, you will be prompted for one.
- `ase_service_name` is the name of the Adaptive Server Enterprise specified in the `interfaces`, `sql.ini`, or LDAP file. There is no default for this parameter value. If you do not supply a value for this parameter, you will be prompted for one.
- `property_file` is the location and name of the properties file to update. The default location for UNIX is `SYBASE/WS-12_5/props/ws.properties`, or `%SYBASE%\WS-12_5\props\ws.properties` for Windows.
- `-v` specifies that the Consumer component should display version information on start-up.

You can start both the Producer and Consumer components for Adaptive Server Enterprise Web Services by executing the `runws` script, which is located in the `bin` directory:

```
runws -U <ase_username> -P <ase_password> -S
```



```
<ase_service_name> -f <property_file> -v
```

where the parameter values are the same as for the *runproducer* script.

Conditions

The Web Services Consumer component will start if the following conditions are met:

- The *ase_service_name* provided is located on an LDAP server pointed to by the *libtcl.cfg* file or in the *interfaces* file for Adaptive Server Enterprise.

Adaptive Server Enterprise Web Services first searches for an entry containing the value of *ase_service_name* on an LDAP server pointed to by the *libtcl.cfg* file. Adaptive Server Enterprise Web Services locates the *libtcl.cfg* file using the `com.sybase.ase.ws.libtcl` entry in the *ws.properties* file. If no entry is found on an LDAP server, Adaptive Server Enterprise Web Services looks for an entry in the *interfaces* file for Adaptive Server Enterprise.

Note On Windows systems, the *interfaces* file is named *sql.ini*.

Adaptive Server Enterprise Web Services locates the *interfaces* file using the `com.sybase.ase.ws.interfaces` entry in the *ws.properties* file.

- A successful login can be made using the *ase_username* and *ase_password* provided.

Note The password for an Adaptive Server Enterprise user may be set to a null string. For null passwords on UNIX, use the token `nopasswordspecified` anywhere the password is required. On Windows, you can use the empty string, “”. Sybase does not recommend using null passwords for security.

- The login to Adaptive Server Enterprise has sa role privileges.
- The following stored procedure command has been executed in isql for your Adaptive Server Enterprise:

```
sp_configure 'enable webservices', 1
```

Verification

After successfully executing the *runconsumer* script, verify that Adaptive Server Enterprise Web Services is enabled and that the Web Services Consumer component is running.

❖ Verifying that Adaptive Server Enterprise Web Services is enabled

To verify that Adaptive Server Enterprise Web Services is enabled:

- Execute the following command on Adaptive Server Enterprise:

```
sp_configure 'enable webservices'
```

If *sp_configure* returns a value of 1, the Web Services feature has been enabled. A return value of 0 indicates the feature is not enabled.

❖ Verifying that the Web Services Consumer component is running

- Check the *consumer.log* file in the *logs* directory for messages indicating that the Web Services Consumer component is running. For example:

```
2004-03-29 16:29:26.047 INFO [main] - Consumer listening on port: 8183
```

Stopping the Web Services Consumer

To stop the Web Services Consumer component for Adaptive Server Enterprise Web Services, execute the *stopconsumer* script, which is located in the *bin* directory:

```
stopconsumer -U <ase_username> -P <ase_password> -S  
<ase_service_name> -f <property_file> -v
```

where:

- *ase_username* is the user name for the Adaptive Server Enterprise. There is no default for this parameter value. If you do not supply a value for this parameter, you will be prompted for one.
- *ase_password* is the password for the Adaptive Server Enterprise. There is no default for this parameter value. If you do not supply a value for this parameter, you will be prompted for one.
- *ase_service_name* is the name of the Adaptive Server Enterprise specified in the *interfaces*, *sql.ini*, or LDAP file. There is no default for this parameter value. If you do not supply a value for this parameter, you will be prompted for one.

- *property_file* is the location and name of the properties file to update. The default location for UNIX is `$SYBASE/WS-12_5/props/ws.properties`, or `%SYBASE%\WS-12_5\props\ws.properties` for Windows.

You can stop both the Producer and Consumer components for Adaptive Server Enterprise Web Services by executing the *stopws* script, which is located in the *bin* directory:

```
stopws -U <ase_username> -P <ase_password> -S  
<ase_service_name> -f <property_file> -v
```

where the parameter values are the same as for the *stopconsumer* script.

Conditions

The Web Services Consumer component will stop if the following conditions are met:

- The *ase_service_name* provided is located on an LDAP server pointed to by the *libtcl.cfg* file or in the *interfaces* file for Adaptive Server Enterprise.

Adaptive Server Enterprise Web Services first searches for an entry containing the value of *ase_service_name* on an LDAP server pointed to by the *libtcl.cfg* file. Adaptive Server Enterprise Web Services locates the *libtcl.cfg* file using the `com.sybase.ase.ws.libtcl` entry in the *ws.properties* file. If no entry is found on an LDAP server, Adaptive Server Enterprise Web Services looks for an entry in the *interfaces* file for Adaptive Server Enterprise.

Note On Windows systems, the *interfaces* file is named *sql.ini*.

Adaptive Server Enterprise Web Services locates the *interfaces* file using the `com.sybase.ase.ws.interfaces` entry in the *ws.properties* file.

- A successful login can be made using the *ase_username* and *ase_password* provided.

Note The password for an Adaptive Server Enterprise user may be set to a null string. For null passwords on UNIX, use the token `nopasswordspecified` anywhere the password is required. On Windows, you can use the empty string, `""`. Sybase does not recommend using null passwords for security.

- The login to Adaptive Server Enterprise has `sa` role privileges.

Using sp_webservices

The `sp_webservices` stored procedure creates and manages the proxy tables used in the Consumer component of Web Services. This section documents the options and parameters for `sp_webservices`.

The `sp_webservices` stored procedure has the following options:

- `add` – creates a proxy table.
- `list` – lists the proxy tables mapped to a WSDL file.
- `modify` – modifies timeout setting.
- `remove` – removes proxy tables mapped to a WSDL file.
- `help` – displays usage information for `sp_webservices`.

Note For information on restrictions for Web Services proxy tables, see “Web Services proxy table restrictions” in Chapter 7, “Troubleshooting.”

add

The `add` option is used to create a proxy table for a Web method specified by a WSDL file. When the `add` option is used successfully, the `list` option is invoked automatically to describe the schema of the new proxy table.

Syntax

```
sp_webservices 'add', 'wsdl_uri' [, sds_name]
[, 'web_method=proxy_table
  [,web_method=proxy_table ]* ' ]
```

Parameters

- *wsdluri*
The location for the WSDL file to be mapped to the new proxy table. If this parameter is specified, Web Services ensures that the URI exists in the `syswsdl` table.
- *sds_name*
The name specified for the Web Services Consumer in the *interfaces* or *sql.ini* file. The default value is `ws`. If no entry exists in the `sysattributes` table, an error results.
- *method_name*
The name of the Web method to be mapped to a proxy table. The *method_name* specified must be the name of a Web method specified in the associated WSDL file.

- *proxy_table*

The name of proxy table to which the Web method specified in *method_name* is mapped.

Usage

If you not specify *method_name* and *proxy_table* values for a Web method, the proxy table generated for the Web method is, by default, the name of the Web method specified in the WSDL file. If there is already a proxy table with the name of this Web method, a new proxy table is generated with a name like the following:

method_nameN

where *method_name* is the default proxy table name, and N is a digit from 1 to 9 denoting each successive mapping of the Web method.

If you do specify *method_name* and *proxy_table* values for a Web method, the name of the proxy table must be new. If there is already a proxy table with the name specified in *proxy_table*, an error results, and none of the Web methods specified in the add option are mapped to proxy tables.

The output from the add option lists the methods that have been successfully mapped to proxy tables as well as those that have not been mapped. The name of a proxy table for an unmapped Web method is indicated as NULL.

list

The list option is used to list Web methods described in a WSDL file.

Syntax

```
sp_webservices 'list' [, 'wsdl_uri'] [, sds_name]
```

Parameters

- *wsdluri*

The URI for the mapped WSDL file. If no value is specified for *wsdluri*, the list option displays information about all Web methods that have been mapped to proxy tables.

- *sds_name*

The name of the SDS server specified for the Consumer component in the *interfaces* or *sql.ini* file. The default value is ws. If no entry exists in the *sysattributes* table, an error results.

If neither the *wsdluri* nor the *sds* parameter is specified, all entries in the *sysattributes* table are listed, ordered by *wsdlid*.

Usage If the Web methods described in the WSDL file have already been mapped to proxy tables, the list option prints information about each proxy table. If the Web methods described in the WSDL file have not already been mapped to proxy tables, the list option prints SQL that can be used to create proxy tables.

modify

The modify option is used to modify the attribute information for a WSDL file.

Syntax `sp_webservices 'modify', 'wsdluri', 'timeout=time'`

Parameters

- *wsdluri*
The URI of the WSDL file for which attribute information is to be changed.
- *time*
The interval in seconds during which a Web method must respond before the operation is aborted.

remove

The remove option is used to remove a proxy table mapping for a Web method.

Syntax `sp_webservices 'remove', 'wsdluri' [, sds_name]`

Parameters

- *wsdluri*
The URI of the WSDL file for which the proxy table is to be removed.
- *sds_name*
The name of the SDS server specified for the Consumer component in the *interfaces* or *sql.ini* file. The default value is ws. If no entry exists in the sysattributes table, an error results.

help

The help option provides instructions and examples illustrating how to use the sp_webservices stored procedure.

Syntax `sp_webservices help [, 'option']`

Parameters

- *option*
The option for which to provide detailed instructions. Valid values are add, list, remove, and modify.

Usage If no value is specified for *option*, the help option prints a brief syntax description for the add, list, remove, and modify options.

Invoking a Web service

To invoke a Web service using the Web Services Consumer, use the following procedure:

❖ Invoking a Web service

- 1 Start the Web Services Consumer.
- 2 Use the add option of `sp_webservices` to map the Web service to a proxy table in Adaptive Server Enterprise.
- 3 Use `sp_help` to determine the input and output parameters needed to invoke the Web method.
- 4 Invoke the Web method with a select statement on the proxy table.

Example 1

This example invokes an RPC/encoded Web method to display stock information.

Use the add option of `sp_webservices` to map Web methods to proxy tables:

```
1>sp_webservices "add" ,
"http://webservices.try.sybase.com/web/services/StockQuotes?wsdl" , ws
2>go
```

Two Web methods are mapped to proxy tables. One of these Web methods is named `getQuote`.

Invoke the Web method by selecting from the proxy table:

```
1> SELECT *
FROM getQuote
WHERE _symbol = 'SY'
2> go

getQuoteReturn      _symbol
-----
22.480000           SY
(1 row affected)
```

Example 2

This example invokes a Web method to display stock information within an XML document.

Use the add option of sp_webservices to map Web methods to proxy tables:

```
1>sp_webservices "add" ,
"http://www.xignite.com/xquotes.asmx?WSDL" , ws
2>go
```

Fifteen Web methods are mapped to proxy tables. One of these Web methods is named GetQuotes.

Invoke the Web method by selecting the outxml column of the GetQuotes proxy table:

```
1> SELECT    outxml
FROM        GetQuotes
WHERE       _inxml  = '<GetQuotes
xmlns="http://www.xignite.com/services/"
<Symbol>SY</Symbol></GetQuotes>'
2> go

outxml

----- ...
<?xml version="1.0" encoding="UTF-8" ?>
<GetQuotesResponse xmlns= ....>
...
</GetQuotesResponse>
(1 row affected)
```

Example 3

This example invokes a Web method to display stock information with a join between the method proxy table and another table. To use this Web service, you must create a table to hold symbols representing stocks:

```
1> create table stocksymbol(symbol varchar(100))
2> go
```

Insert data into the stocksymbol table:

```
1> insert stocksymbol values("SY")
2> insert stocksymbol values("PSFT")
3> insert stocksymbol values("ORCL")
4> insert stocksymbol values("MSFT")
5> insert stocksymbol values("SUNW")
```



```
6> go
```

Use the add option of `sp_webservices` to map Web methods to proxy tables:

```
1>sp_webservices "add" ,  
"http://webservices.try.sybase.com/web/services/StockQuotes?wsdl" , ws  
2>go
```

Two Web methods are mapped to proxy tables. One of these Web methods is named `getQuote`.

Invoke the Web method by executing a join on the `symbol` column of the `stocksymbol` table and the `_symbol` column of the `getQuote` proxy table:

```
1> SELECT symbol,getQuoteReturn  
FROM getQuote , stocksymbol  
WHERE getQuote._symbol = stocksymbol.symbol  
2> go
```

symbol	getQuoteReturn
-----	-----
SY	22.469999
PSFT	21.040001
ORCL	13.730000
MSFT	27.070000
SUNW	5.700000

(5 rows affected)

Example 4

This example invokes a Web method through a view to display stock information.

Note This example requires the ASE XML Management Option Package.

To use this Web service, you must create a table to hold symbols representing stocks:

```
1> create table stocksymbol(symbol varchar(100))  
2> go
```

Insert data into the `stocksymbol` table:

```
1> insert stocksymbol values("SY")  
2> insert stocksymbol values("PSFT")  
3> insert stocksymbol values("ORCL")  
4> insert stocksymbol values("MSFT")
```

```
5> insert stocksymbol values("SUNW")
6> go
```

Use the add option of sp_webservices to map Web methods to proxy tables:

```
1>sp_webservices "add" ,
"http://www.xignite.com/xquotes.asmx?WSDL" , ws
2>go
```

Fifteen Web methods are mapped to proxy tables. One of these Web methods is named GetQuotes.

Now create a view that invokes the GetQuotes Web method:

```
1> CREATE VIEW getstockvw as
SELECT
    Symbol = xmlextract('//Quote/Symbol/text()',outxml
    returns varchar(10)),
    Name = xmlextract('//Quote/Name/text()',outxml
    returns varchar(100)),
    Date = xmlextract('//Quote/Date/text()',outxml
    returns date),
    Time = xmlextract('//Quote/Time/text()',outxml
    returns varchar(25)),
    Openval = xmlextract('//Quote/Open/text()',outxml
    returns decimal(15,3)),
    High = xmlextract('//Quote/High/text()',outxml
    returns decimal(15,3)),
    Low = xmlextract('//Quote/Low/text()',outxml returns
    decimal(15,3)),
    Last = xmlextract('//Quote/Last/text()',outxml
    returns decimal(15,3)),
    Volume = xmlextract('//Quote/Volume/text()',outxml
    returns decimal(15,3)),
    PercentChange =
    xmlextract('//Quote/PercentChange/text()',outxml
    returns decimal(15,3))
FROM      GetQuotes ,stocksymbol
WHERE     _inxml = '<GetQuotes
            xmlns="http://www.xignite.com/services/">
            <Symbol>'+symbol+'</Symbol>
            </GetQuotes>'
```

```
2> go
```

Select from the getstockvw view to view output from the GetQuotes method:

```
1> select * from getstockvw where Symbol in
('SY', 'ORCL')
2> go
```

```

Symbol Name          Date          Time  O
penval High  Low  Last  Volume  PercentChange
-----
-----
SY      SYBASE INC      Feb 11 2004 2:55PM ET
22.22  22.53  22.03  22.53  344600.00  1.40
ORCL    ORACLE CORP      Feb 11 2004 3:00PM ET
13.53  13.79  13.33  13.73  29931784.00  2.54
(2 rows affected)

```

Example 5

This example uses a trigger with a Web service to deliver quotes and other information about a stock in the form of an XML document.

Note This example requires the ASE XML Management Option Package.

To use this Web service, you must create a table to hold stock information:

```

1> create table stocktab(symbol varchar(25), value
numeric(10,2) null, xmldoc text null)
2> go

```

Use the add option of `sp_webservices` to map Web methods to proxy tables:

```

1>sp_webservices "add",
"http://www.xignite.com/xquotes.asmx?WSDL", ws
2>go

```

Fifteen Web methods are mapped to proxy tables. One of these Web methods is named `GetQuotes`.

Now create a trigger that invokes the `GetQuotes` Web method. The trigger places the output from `GetQuotes`, a stock quote for the specified symbol and a summary of other data encapsulated in an XML document, into the `value` and `xmldoc` columns of the `stocktab` table:

```

1> CREATE trigger stocktrig ON stocktab FOR
insert,update
AS
UPDATE stocktab
SET    stocktab.value =
      xmlextract('//Quote/Last/text()',outxml returns
      numeric(10,2)), xmldoc = outxml
FROM  stocktab, inserted , GetQuotes
WHERE _inxml  = '<GetQuotes

```

```
xmlns="http://www.xignite.com/services/">
<Symbol>'+stocktab.symbol+'</Symbol>
</GetQuotes>'
AND stocktab.symbol = inserted.symbol
2> go
```

This trigger executes when data is inserted into the stocktab table:

```
1> insert stocktab values('SY',null,null)
2> go
1> insert stocktab values('PSFT',null,null)
2> go
1> insert stocktab values('MSFT',null,null)
2> go
```

View the updated data with a select on the stocktab table:

```
1> select value, xmldoc from stocktab
2> go
```

```
value          xmldoc
-----
22.53          <?xml version="1.0" encoding="UTF-8" ?>
<GetQuotesResponse ...> ...</GetQuotesResponse>
21.04          <?xml version="1.0" encoding="UTF-8" ?>
<GetQuotesResponse ...> ...</GetQuotesResponse>
27.07          <?xml version="1.0" encoding="UTF-8" ?>
<GetQuotesResponse ...> ...</GetQuotesResponse>
(3 rows affected)
```

Adaptive Server Enterprise Web Services Logging

By default, Adaptive Server Enterprise Web Services logs only informational and error messages. For details on how to log more detailed information, contact Sybase Technical Support.

This section concerns the following topics:

- Adaptive Server Enterprise Web Services log files
- Rolling over log files

Adaptive Server Enterprise Web Services log files

Adaptive Server Enterprise Web Services logs activity into three files:

- `consumer.log`
- `producer.log`
- `http.log`

Logging is implemented in Adaptive Server Enterprise Web Services using the Apache log4j framework. For information on log4j, refer to the following Web documentation: <http://jakarta.apache.org/log4j/docs/>.

consumer.log

The *consumer.log* file contains all information and error messages from a Web Services Consumer.

producer.log

The *producer.log* file contains all information and error messages from a Web Services Producer.

http.log

The *http.log* file holds all HTTP requests in NCSA Request Log format. An HTTP request exists for each Web method invoked.

Rolling over log files

Logging is implemented in Adaptive Server Enterprise Web Services using the Apache log4j framework. For information on specific log4j parameters and implementing a rollover policy, refer to the following Web documentation: <http://jakarta.apache.org/log4j/docs/>.

Sybase Central

The Sybase Central plug-in for Adaptive Server Enterprise Web Services enables you to perform the following tasks:

- Consume a Web service
- Execute a Web service
- Delete a Web service
- Display properties for a Web service

For details, refer to the online help for Adaptive Server Enterprise Web Services in Sybase Central.

Using the Web interface

Adaptive Server Enterprise Web Services provides a Web interface for the following:

- Performing administrative tasks
- Executing Web methods

❖ Starting the Web interface

Use the following procedure to start the Web interface:

- 1 Make sure Adaptive Server Enterprise is running.
- 2 Start the Producer component by executing the *runproducer* or *runws* script.
- 3 Open a Web browser, and go to the following URL:

`https://domainhostname:port`

where:

- *domainhostname* is the name of the machine on which your Adaptive Server Enterprise has been installed.
- *port* is the port on which the Producer component listens for SSL requests.

Note You must have SSL configured to access Web Services through the Web interface.

Performing administrative tasks

Using the Web interface, you can configure the Producer component and view Web Services logs.

Configuring the Producer

To configure the Producer component, use the following procedure.

❖ Configuring the Producer component

- 1 Click the Producer link in the Administration menu of the Web interface. The Web interface displays the Producer component properties and their current settings.
- 2 Provide updated values for the properties displayed.
- 3 Click Save.

Viewing log files

To view the *producer.log*, *consumer.log*, or *http.log* file, select the links for Producer, Consumer, or HTTP, respectively.

Executing Web methods

To execute a Web method using the Web interface, use the following procedure:

❖ Executing a Web method

- 1 Select the Execute link under the Web Methods menu.
- 2 Provide values for the following fields:
 - Username – The user name used to connect to Adaptive Server Enterprise.
 - Password – The password used to connect to Adaptive Server Enterprise.
 - ASE Service – The name of the Web service to be invoked.
 - SQL – SQL arguments for the Web service to be invoked.
 - SQLX Options – SQLX options for the Web Service to be invoked.

- Output – Output format. This is a drop-down list.
- 3 Click Submit to execute the Web method.

Tools are provided under the *samples* directory to create and run sample clients. This chapter documents the following sample applications provided for the Web Services Producer component:

Topic	Page
Apache sample client	59
Microsoft .NET sample client	64

Apache sample client

This section describes the sample client and script found in the `$$SYBASE/WS-12_5/samples/apacheclient` directory in UNIX, or in the `%SYBASE%\WS-12_5\samples\apacheclient` directory in Windows.

Note If you intend to run the Apache sample client on a machine other than the one on which Adaptive Server Enterprise Web Services is installed, you must copy the contents of the `/apacheclient/lib` directory to that machine.

Creating the sample client

To use the sample script provided, you must first create the sample client.

❖ Creating the sample client

- Make sure the JRE variable points to your JRE by changing, if necessary, the variable definitions in all scripts in the *apacheclient* directory. You must use JRE version 1.3.1 or later. By default, the JRE supplied in the UNIX `$$SYBASE_JRE` or Windows `%SYBASE_JRE%` directory is used.

Once you have created an Adaptive Server Enterprise Web Services client, you can run the sample script to execute stored procedures and SQL statements. This script can be found in the *apacheclient* directory.

runexecute

The *runexecute* script executes a stored procedure or Transact-SQL statement on Adaptive Server Enterprise through Adaptive Server Enterprise Web Services. This sample application invokes the *execute* Web method.

Syntax

```
runexecute "web_service_URL" aseServerName user_ID
password "SQLX_option" output_class count
"sql_statement"
```

Parameters

- *web_service_URL*
The location of the Web service being used.
- *aseServerName*
SOAP string indicating the name of the Adaptive Server Enterprise server in the *interfaces* file or LDAP server.
- *user_ID*
The user ID needed to log in to the Adaptive Server Enterprise.
- *password*
The password needed to log in to the Adaptive Server Enterprise.
- *SQLX_option*
String indicating one or more option parameters. These parameters specify characteristics of the SQLX result set. The following are valid option parameters:
 - `binary={hex | base64}`
 - `columnstyle={element | attribute}`
 - `format={yes | no}`
 - `header={yes | no}`
 - `nullstyle={attribute | omit}`
 - `prefix="value"`
 - `root={yes | no}`

- `rowname="value"`
- `schemaloc="value"`
- `statement={yes | no}`
- `tablename="value"`
- `targetns="value"`

You must provide values for *value*. For more information on SQLX functions and options, see *XML Services in Adaptive Server Enterprise*.

- *output_class*

The kind of output desired. The following are valid values for this parameter:

- `schema` – returns an XML schema.
- `dtd` – returns an XML DTD.
- `data` – returns a result set.
- `all` – returns schema, DTD, and data.

- *count*

The number of times to execute the statement. If the value of `count` is greater than 1, a session is created, and a persistent connection is used.

- *sql_statement*

The statement to be executed on Adaptive Server Enterprise. This statement must be delimited by double quotes.

Example 1

This example checks the version number for Adaptive Server Enterprise using a `select` statement.

```
runexecute "http://johndoe-sun:8183/services/ase"
johndoe-sun sa nopasswordspecified "tablename=ws" all 1
"select @@version"
```

Adaptive Server Enterprise Web Services returns an XML schema, a DTD, and a result set containing the result of the executed statement.

Example 2

This example executes a stored procedure called `booksales` on the `pubs2` database. The stored procedure returns the number of copies sold for a specified book title ID.

```
runexecute "http://johndoe-sun:8183/services/ase"
johndoe-sun sa nopasswordspecified
"columnstyle=attribute,format=no,rowname=wsrow,prefix=
```

```
Unnamedcol,nullstyle=attribute,header=yes" all 1
"execute booksales MC2222"
```

Adaptive Server Enterprise Web Services returns an XML schema, a DTD, and a result set containing the result of the executed statement.

This is the result set returned:

```
<?xml version="1.0" ?>
<resultset
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<wsrow title="Silicon Valley Gastronomic Treats"
total_sales="2032" Unnamedcol1="Books sold"/>
</resultset>
```

This is the DTD returned:

```
<!DOCTYPE ws [
<!ELEMENT resultset (row*)>
<!ELEMENT row (title, total_sales, Unnamedcol1)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT total_sales (#PCDATA)>
<!ELEMENT Unnamedcol1 (#PCDATA)>
]>
```

This is the schema returned:

```
<?xml version="1.0" ?>
<xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:sqlxml="http://www.iso-standards.org/mra/9075/
sqlxml">
<xsd:import
namespace="http://www.w3.org/2001/XMLSchema"
schemaLocation="http://www.iso-standards.org/mra/
9075/sqlxml.xsd" />
<xsd:complexType
name="RowType.resultset">
<xsd:attribute name="title"
type="VARCHAR_80" use="required"/>
<xsd:attribute name="total_sales" type="INTEGER"
use="optional"/>
<xsd:attribute name="Unnamedcol1"
type="VARCHAR_24" use="optional"/>
</xsd:complexType>
<xsd:complexType
name="TableType.resultset">
<xsd:sequence>
<xsd:element name="wsrow"
```

```

        type="RowType.resultset" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="VARCHAR_80">
    <xsd:restriction base="xsd:string">
        <xsd:length value="80"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="INTEGER">
    <xsd:restriction base="xsd:integer">
        <xsd:maxInclusive value="2147483647"/>
        <xsd:minInclusive value="-2147483648"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="VARCHAR_24">
    <xsd:restriction base="xsd:string">
        <xsd:length value="24"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:element name="resultset"
    type="TableType.resultset"/>
</xsd:schema>

```

Example 3

This example executes a SQL query on the pubs2 database. The query returns the last names and cities of residence for authors who do not live in the same city as their publisher.

```

runexecute "http://johndoe-sun:8183/services/ase"
johndoe-sun sa nopasswordspecified
"tablename=ws,header=yes,schemaloc='http://www-
edm/remote/svr/xmltestdir/resultset.xsd',targetns='htt
p://www-edm/remote/svr/xmltestdir/'" data 1 "select
distinct au_lname, authors.city from publishers,
authors where authors.city not in (select city from
publishers where authors.city = publishers.city) "

```

Adaptive Server Enterprise Web Services returns a result set containing the result of the executed statement.

Microsoft .NET sample client

This section describes the sample client and script found in the `$$SYBASE/WS-12_5/samples/ms.net/Execute/bin/Release` directory in UNIX, or in the `%SYBASE%\WS-12_5\samples\ms.net\Execute\bin\Release` directory in Windows.

Downloads for Microsoft .NET can be found at the following URL:

<http://msdn.microsoft.com/library/default.asp?url=/downloads/list/netdevframework.asp>

Note This URL is current as of the date of publication for this document but may change over time.

Creating the sample client

To use the sample script provided, you must first create the sample client. Once you have created the sample client, you can run the sample script documented in this section. This script can be found in the *Release* directory.

Execute.exe

Execute.exe executes a stored procedure or Transact-SQL statement on Adaptive Server Enterprise through Adaptive Server Enterprise Web Services. This sample application invokes the `execute` Web method.

Syntax

```
Execute.exe "web_service_URL" aseServerName user_ID  
password "SQLX_option" output_class count  
"sql_statement"
```

- Parameters
- *web_service_URL*
The location of the Web service being used.
 - *aseServerName*
SOAP string indicating the name of the Adaptive Server Enterprise server in the *sql.ini* file or LDAP server.
 - *user_ID*
The user ID needed to log in to the Adaptive Server Enterprise.

- *password*
The password needed to log in to the Adaptive Server Enterprise.
- *SQLX_option*
String indicating one or more option parameters. These parameters specify characteristics of the SQLX result set. The following are valid option parameters:
 - `binary={hex | base64}`
 - `columnstyle={element | attribute}`
 - `format={yes | no}`
 - `header={yes | no}`
 - `nullstyle={attribute | omit}`
 - `prefix="value"`
 - `root={yes | no}`
 - `rowname="value"`
 - `schemaloc="value"`
 - `statement={yes | no}`
 - `tablename="value"`
 - `targetns="value"`

You must provide values for *value*. For more information on SQLX functions and options, see *XML Services in Adaptive Server Enterprise*.
- *output_class*
The kind of output desired. The following are valid values for this parameter:
 - `schema` - returns an XML schema.
 - `dtd` - returns an XML DTD.
 - `data` - returns a result set.
 - `all` - returns schema, DTD, and data.
- *count*
The number of times to execute.
- *sql_statement*

The statement to be executed on Adaptive Server Enterprise. This statement must be delimited by double quotes.

Example 1

This example checks the version number for Adaptive Server Enterprise.

```
Execute.exe "http://johndoe-sun:8183/services/ase"  
johndoe-sun sa nopasswordspecified "tablename=ws" all 1  
"select @@version"
```

Adaptive Server Enterprise Web Services returns an XML schema, a DTD, and a result set containing the result of the executed statement.

Example 2

This example executes a stored procedure called booksales on the pubs2 database. The stored procedure returns the number of copies sold for a specified book title ID.

```
Execute.exe "http://johndoe-sun:8183/services/ase"  
johndoe-sun sa nopasswordspecified  
"columnstyle=attribute,format=no,rowname=wsrow,prefix=  
Unnamedcol,nullstyle=attribute,header=yes" all 1  
"execute booksales MC2222"
```

Adaptive Server Enterprise Web Services returns an XML schema, a DTD, and a result set containing the result of the executed statement.

Example 3

This example executes a SQL query on the pubs2 database. The query returns the last names and cities of residence for authors who do not live in the same city as their publisher.

```
Execute.exe "http://johndoe-sun:8183/services/ase"  
johndoe-sun sa nopasswordspecified  
"tablename=ws,header=yes,schemaloc='http://www-  
edm/remote/svr/xmltestdir/resultset.xsd',targetns='htt  
p://www-edm/remote/svr/xmltestdir/'" data 1 "select  
distinct au_lname, authors.city from publishers,  
authors where authors.city not in (select city from  
publishers where authors.city = publishers.city)"
```

Adaptive Server Enterprise Web Services returns a result set containing the result of the executed statement.

Troubleshooting

This section describes troubleshooting procedures and error messages.

This chapter provides troubleshooting information on the following topics:

Topic	Page
Troubleshooting issues	69
Diagnostic tools	75
Messages	75

Troubleshooting issues

The following are issues that can assist you in troubleshooting Adaptive Server Enterprise Web Services.

HTTP-authorized Web methods unusable

Issue

The Consumer component cannot complete an HTTP authentication handshake and cannot consume Web methods that use this form of authentication.

User action

Sybase recommends that you use a stronger form of authentication than HTTP authentication, such as SSL.

Encoding issue for roman8 character set on HP

Issue

When you use the Producer or Consumer component, the session used to connect to Adaptive Server Enterprise uses the iso_1 character set due to a limitation with the JVM. This occurs for Adaptive Server Enterprise running on HP platforms, where the default character set is roman8, which does not have a valid mapping.

User action

None.

Unmapped RPC/encoded Web method

Issue	If an RPC/encoded Web method has no input or output parameters, it cannot be mapped to a proxy table. A proxy table for a Web method without parameters would have no columns. A table with no columns cannot be created in Adaptive Server Enterprise.
User action	Modify the Web method to include an input or output parameter.

Truncated document/literal results

Issue	If a Web service returns more data than the value of the @@textsize global variable, the data is truncated to the size specified by @@textsize. Consequently, the data returned may not form a valid XML document.
User action	No error is raised because this is the expected behavior for Adaptive Server Enterprise when text or image data is truncated with a select command. However, a warning is logged for the Consumer component, so check your <i>consumer.log</i> file.

Starting the Producer and Consumer

Issue	The <i>runproducer</i> , <i>runconsumer</i> , or <i>runws</i> script does not successfully start a Web Services component.
User action	<ol style="list-style-type: none">1 Make sure the port you are attempting to use is not already in use by another process.2 Make sure you have the correct JRE installed. Adaptive Server Enterprise Web Services requires JRE 1.3.1 or later. To check your JRE version, enter the following at your command prompt: <pre>java -version</pre>3 If you want Web Services to run with a properties file other than <i>ws.properties</i>, you must specify the absolute path for the file. For example, to run the Producer component with a different properties file: <pre>C:\sybase\WS-12_5\bin\runproducer -f C:\sybase\WS-12_5\props\myfile.properties</pre>
Issue	The Web Services Producer component finds the specified <i>ase_service_name</i> in the <i>interfaces</i> file, but the <i>producer.log</i> shows the following error messages: <pre>ERROR [main] - Error locating libtcl.cfg file.</pre>

```
ERROR [main] - java.io.FileNotFoundException: LDAP
config File does not exist
```

Locating WSDL

Issue	A client connecting to the Web Services Producer through a Web browser cannot find the WSDL file, or the Web Services Consumer cannot find the WSDL file to perform proxy-table mapping.
User action	Verify that the Web Services component is running. If using the Web Services Producer through a browser, make sure the browser URL indicates <code>https://</code> for an SSL connection and <code>http://</code> for a standard connection.

Specifying entries in ws.properties

Issue	Because the backslash “\” symbol is used as an escape character, entries that use single backslash symbols are not interpreted correctly. For example:
-------	--------------------------------------------------------------------------------------------------------------------------------------------------------

```
com.sybase.ase.ws.interfaces = d:\sybase\ini\sql.ini
```

User action	Escape the backslash with another backslash. For example:
-------------	-----------------------------------------------------------

```
com.sybase.ase.ws.interfaces = d:\\sybase\\ini\\sql.ini
```

You can also use forward slashes. For example:

```
com.sybase.ase.ws.interfaces = d:/sybase/ini/sql.ini
```

Windows NT command-line arguments

Issue	Scripts do not run on Windows NT when no space is placed between arguments and argument values. For example, the following invocation of the <i>configssl</i> script will not execute the script:
-------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

```
configssl -dhostname
```

User action	Place a space between an argument and its value. For example:
-------------	---------------------------------------------------------------

```
configssl -d hostname
```

Run or stop scripts fail

Issue The *runproducer*, *runconsumer*, *runws*, *stopproducer*, *stopconsumer*, or *stopws* script fails to execute.

User action If any of these scripts fail to execute, do the following:

- Verify that your Adaptive Server Enterprise is running.
- Make sure that the user name and password you specified are valid to log in to your Adaptive Server Enterprise.
- Check the *producer.log* or *consumer.log* file for any error messages.
- Verify that the *ase_service_name* provided can be found on an LDAP server pointed to by the *libtcl.cfg* file or in the *interfaces* file for Adaptive Server Enterprise.

Note On Windows systems, the *interfaces* file is named *sql.ini*.

- Verify that the username has *sa_role* privileges.

runexecute or *execute* fails

Issue The *runexecute* sample application or the *execute* method fails to execute.

User action If the *runexecute* sample application or the *execute* method fails to execute, do the following:

- Verify that your Adaptive Server Enterprise is running.
- Make sure that your Adaptive Server Enterprise has a valid ASE_WEBSERVICE license in SySAM.
- Make sure that Adaptive Server Enterprise Web Services has been enabled by entering the following command on your Adaptive Server Enterprise:

```
sp_configure 'enable webservices', 1
```

If Adaptive Server Enterprise Web Services is enabled, an informational message indicating this is added to the *producer.log* file. If Adaptive Server Enterprise Web Services is not enabled, a warning message indicating this is added to the *producer.log* file.

Null passwords

- Issue** The password for an Adaptive Server Enterprise user may be set to a null string.
- User action** On UNIX, use the token `nopasswordspecified` anywhere the password is required, including the `runproducer`, `runconsumer`, `stopproducer`, `stopconsumer`, `runws` and `stopws` scripts.
- On Windows, the empty string, "", can be used.

Specifying SOAP endpoints with SSL

- Issue** Adaptive Server Enterprise Web Services methods or sample applications do not return results with the `aseServerName` specified at invocation.
- User action** Make sure the `aseServerName` name is a valid SOAP endpoint. If you are using a DNS alias, make sure the alias resolves to a valid SOAP endpoint. If you are using SSL, make sure the endpoint specified by `aseServerName` is the same name you supplied in creating an SSL certificate with the `configssl` script. For example:

```
configssl -d mydomainhostname -h 8182
```

Here, the value of `aseServerName` supplied when invoking an Adaptive Server Enterprise Web Services method or sample application must be "https://mydomainhostname:8182". The method or sample application will not return results if you substitute "localhost" or an IP address for `aseServerName` when using SSL.

Abnormal termination of sp_webservices 'add'

- Issue** Proxy tables created during execution of the `sp_webservices 'add'` option remain after an abnormal termination of `sp_webservices`, as with a Ctrl+C interrupt or an Adaptive Server Enterprise crash.
- User action** Use the `remove` option to delete any proxy tables created by the `add` option when `sp_webservices` terminated abnormally.

Web Services proxy table restrictions

Because they point to Web methods, Web Services proxy tables are of type procedure and are subject to all restrictions for this type:

- *Commands* – You cannot issue a delete, update, or insert command against a table of type procedure, nor can you issue a create index, truncate table, or alter table command.
- *Joins* – A Web Services proxy table can only be joined with one other table, and that table cannot be another Web Services proxy table.
- *Queries* – Column names beginning with underscore ('_') are used to specify parameters, which are not part of the result set for the remote procedure. These columns are referred to as parameter columns and must be in the where clause of a select statement.

For a complete description of the restrictions for procedure tables, see the *Component Integration Services User's Guide*.

Issue

A proxy table with the same name as a Web Services proxy table generated through execution of the add option of `sp_webservices` may be erroneously designated as a Web Services proxy table. Joins are processed for Web Services proxy tables differently than joins for regular proxy tables.

User action

Confine Web Services proxy tables to a unique database.

Issue

The reserved word `or` is not supported in the where clause for a query involving Web Services proxy tables. This may result in errors for queries that are translated internally into queries that use the reserved word `or`. For example, the following query selects from the Web Services proxy table `testint`:

```
select * from testint where _in0 in (1, 2)
```

This query is translated internally into the following:

```
select * from testint where _in0 = 1 or _in0 = 2
```

Because the internal translation uses the reserved word `or`, the user-submitted query results in an error.

User action

Alter the query so that its internal translation does not use the reserved word `or`. For example, the user-submitted query just described can be altered with the use of temporary tables:

```
create table a (col int)
insert into a values (1)
insert into a values (2)
select * from testint where _in0 in (select col from a)
```


sysattributes table entry

Issue	<p>The following error occurs when <code>sp_webservices 'add'</code> is executed:</p> <pre>Msg 5629, Level 16, State 1: Line 1: Cannot start a remote distributed transaction participant as the local server is not named. Please contact a user with System Administrator role.</pre>
User action	<p>You must have an entry for your Adaptive Server Enterprise in the <code>sysattributes</code> table. To provide an entry, use the <code>sp_addserver</code> stored procedure:</p> <pre>sp_addserver ase_entry, local, ase_entry</pre> <p>where <code>ase_entry</code> is the local name for your Adaptive Server Enterprise from your <code>interfaces</code> or <code>sql.ini</code> file.</p>

Diagnostic tools

Adaptive Server Enterprise Web Services provides JDBC-level tracing to help identify connectivity problems.

Enabling JDBC-level tracing

To enable JDBC-level tracing, refer to the appropriate Java documentation.

Messages

Table 7-1 lists error messages for the `sp_webservices` stored procedure.

Table 7-1: sp_webservices error messages

Message number	Message text
15200	No web methods mapped to proxy tables for the WSDL URI [%s].
15201	WSDL URI in most cases has the suffix ?WSDL. Please verify WSDL URI.
15202	Web Method [%s] not mapped to proxy table because of unsupported datatype.
15203	Received request to execute an unknown procedure [%s].
15204	Caught IOException. This usually indicates an error in communications between ASE and the Consumer.\nDetails: [%s].
15205	Caught SQLException. This usually indicates an error retrieving meta data from ASE.\nDetails: [%s].
15206	Caught an Unknown Exception: Details: [%s].
15207	Caught Remote Web Method Exception (AxisFault). This indicates an error in the remote web method.\nDetails: [%s].
15208	Caught Mapping Exception. This indicates an error in mapping the web method arguments to ASE types.\n Details: [%s].
15209	Caught Service Exception. This usually indicates an incorrect WSDL file.\n Details: [%s].
15210	Received XML input to the webmethod that was is not well formed.
15211	Error in invoking web method (MalformedURLException) Details: [%s].
15212	Caught RemoteException. This usually indicates an error in the network transmission.\n Details: [%s].
15213	Error in invoking web method (Unknown Exception): Details [%s].
15214	Aborting invocation of web method [%s] from proxy table [%s] because the web method expects [%s] arguments and [%s] were received.

Table 7-2 lists informational messages for the sp_webservices stored procedure.

Table 7-2: sp_webservices informational messages

Message number	Message text
19307	Generating proxy tables using sds [%1!] for WSDL URI: [%2!].
19308	Found WSDL Match for [%1!].
19309	The WSDL URI [%1!] was not found in the system.
19310	Updating timeout entries for WSDL URI [%1!] with [%2!].
19311	Removing all web service meta data....
19312	Deleting entries for WSDL URI [%1!].

Message number	Message text
19313	To remove a webservice, a valid SDS server must be supplied. The SDS server [% 1!] was not found. Please use sp_addserver to add the SDS server.
19319	To add a webservice, a valid SDS server must be supplied. \n The SDS server [% 1!] was not found. \nPlease use sp_addserver to add the SDS server.
19320	Verify Consumer is running.
19321	To add a webservice, a valid WSDL URI must be specified.
19322	The WSDL URI [% 1!] specified is already in the system. \n Please use sp_webservices remove first.
19323	To list information about a webservice not loaded in the system, a SDS server must be supplied. The SDS server [% 1!] was not found. \n Please use sp_addserver to add the SDS server.
19324	Must specify a specific wsdl uri to modify.
19325	Must specify something to change for modify.
19326	[% 1!] is not a valid option for sp_webservices modify.
19327	Must specify item=value syntax for modify.

For help information for sp_webservices, enter sp_webservices help at the isql command line.

Appendixes

This section contains the appendixes for this book.



Installation Contents

This appendix describes the contents of the Adaptive Server Enterprise Web Services installation.

Adaptive Server Enterprise Web Services directory tree

Adaptive Server Enterprise Web Services is installed at the same level as the root directory for Adaptive Server Enterprise. The Adaptive Server Enterprise Web Services root directory is named *WS-12_5* and consists of the following subdirectories:

Table A-1: Adaptive Server Enterprise Web Services directories

Directory name	Contents
<i>bin</i>	Scripts for configuring and running Adaptive Server Enterprise Web Services components.
<i>lib</i>	Java libraries and packages used by Adaptive Server Enterprise Web Services.
<i>logs</i>	Default location for log files.
<i>producer</i>	Files and subdirectories for Web Services Producer component at runtime.
<i>props</i>	Files for Adaptive Server Enterprise Web Services properties.
<i>samples</i>	Sample scripts for building and running a sample client with the Web Services Producer component.
<i>scripts</i>	SQL scripts.

bin directory contents

The following table describes the contents of the *bin* directory.

Table A-2: bin directory contents

File/Directory name	Function
<i>configssl</i>	Configures SSL.
<i>getpass.exe</i>	Used for Adaptive Server Enterprise login. This file is only available on Windows.
<i>installws</i>	Installs sp_webservices stored procedure.
<i>runconsumer</i>	Starts Adaptive Server Enterprise Web Services Consumer.
<i>runproducer</i>	Starts Web Services Producer.
<i>runtcpmon</i>	Creates a monitor to trace SOAP messages.
<i>runws</i>	Starts Adaptive Server Enterprise Web Services Consumer and Producer.
<i>stopconsumer</i>	Stops Adaptive Server Enterprise Web Services Consumer
<i>stopproducer</i>	Stops Adaptive Server Enterprise Web Services Producer
<i>stopws</i>	Stops Adaptive Server Enterprise Web Services Consumer and Producer.

Note The Windows versions of these files are suffixed with *.bat*.

***lib* directory contents**

The following table describes the contents of the *lib* directory.

Table A-3: *lib* directory contents

File/Directory name	Function
<i>axis.jar</i>	Apache Axis file.
<i>commons-discovery.jar</i>	Apache Axis file.
<i>commons-logging.jar</i>	Apache Axis file.
<i>flexlm.jar</i>	Library for FLEXlm licensing.
<i>javax.servlet.jar</i>	Servlet library.
<i>jaxrpc.jar</i>	Apache Axis file.
<i>jce1_2_2.jar</i>	JCE file.
<i>jcrt.jar</i>	JSSE library.
<i>jnet.jar</i>	JSSE library.
<i>jsse.jar</i>	JSSE library.
<i>local_policy.jar</i>	JCE file.
<i>log4j-1.2.4.jar</i>	log4j logger.
<i>mail.jar</i>	Apache Axis file.
<i>org.mortbay.jetty-jdk1.2.jar</i>	HTTP server.
<i>saaj.jar</i>	Apache Axis file.
<i>sqlx.jar</i>	SQLX file.
<i>sunjce_provider.jar</i>	JCE file.
<i>tools.jar</i>	Java tools (for example, the javac compiler).
<i>US_export_policy.jar</i>	JCE file.
<i>ws-12.5.jar</i>	Code for Web Services Consumer, Web Services Producer.
<i>ws-12.5d.jar</i>	Debugger Code for Web Services Consumer, Web Services Producer.
<i>wSDL4j.jar</i>	Apache Axis file.
<i>xercesImpl.jar</i>	Xerces parser.
<i>xmlParserAPIs.jar</i>	Xerces parser.

***logs* directory contents**

The following table describes the contents of the *logs* directory.

Table A-4: logs directory contents

File/Directory name	Function
<i>consumer.log</i>	Logs Web Services Consumer component activity.
<i>http.log</i>	Logs Web server activity.
<i>producer.log</i>	Logs Web Services Producer component activity.

producer directory contents

The following table describes the contents of the *producer* directory.

Table A-5: producer directory contents

File/Directory name	Function
<i>keystore</i>	Holds encryption keys.
<i>server</i>	Directory structure required for Jetty.
<i>WEB-INF</i>	Directory structure required for Jetty.
<i>wscertificate.cer</i>	Auto-generated certificate for SSL. This file is present only the <i>configssl</i> script has been run.

props directory contents

The following table describes the contents of the *props* directory.

Table A-6: props directory contents

Directory name	Function
<i>logging.properties</i>	Configuration file for log4j.
<i>ws.properties</i>	All configuration parameters for Web Services Consumer, Web Services Producer.
<i>wsmsg.properties</i>	Configuration file for messages.

samples directory contents

The samples directory contains precompiled and source code for both Apache and Microsoft .NET. You can use source code in your own applications. The following table describes the contents of the *samples* directory.

Table A-7: samples directory contents

Directory name	Function
<i>apacheclient</i>	Directory containing sample scripts for compiling and running the sample client.
<i>ms.net</i>	Samples for .NET.

Configuration Properties

This appendix describes the contents of the Adaptive Server Enterprise Web Services *ws.properties* file and the *myres.properties* file.

ws.properties

The *ws.properties* file contains the following configuration settings for Adaptive Server Enterprise Web Services.

Table B-1: ws.properties entries

<code>com.sybase.ase.ws.consumer.cisport</code>	Indicates the port on which the Web Services Consumer should listen for TDS. The default entry is 8183.
<code>com.sybase.ase.ws.consumer.logfilename</code>	Indicates where the logfile for the Web Services Consumer component should be placed. The default location for UNIX is <code>\$\$SYBASE/WS-12_5/logs/consumer.log</code> , or <code>\$\$SYBASE%\WS-12_5\logs\consumer.log</code> for Windows.
<code>com.sybase.ase.ws.interfaces</code>	Indicates the location of the <i>interfaces</i> or <i>sql.ini</i> file for Adaptive Server Enterprise. The default location for UNIX is <code>\$\$SYBASE/interfaces</code> , or <code>\$\$SYBASE%\ini\sql.ini</code> for Windows.
<code>com.sybase.ase.ws.libtcl</code>	Indicates the location of the <i>libtcl.cfg</i> file used to identify LDAP servers. The default location for 32-bit platforms for UNIX is <code>\$\$SYBASE/config/libtcl.cfg</code> , or <code>\$\$SYBASE%\ini\libtcl.cfg</code> for Windows. The default location for 64-bit platforms for UNIX is <code>\$\$SYBASE/config/libtcl64.cfg</code> , or <code>\$\$SYBASE%\ini\libtcl64.cfg</code> for Windows.
<code>com.sybase.ase.ws.producer.httpport</code>	Indicates the port on which the Web Services Producer should listen for an HTTP connection. The default entry is 8181.

`com.sybase.ase.ws.producer.jettylogfile`

Indicates where the logfile for HTTP requests should be placed. The default location is for UNIX is `$SYBASE/WS-12_5/logs/http.log`, or `%SYBASE%\WS-12_5\logs\http.log` for Windows.

`com.sybase.ase.ws.producer.logfilename`

Indicates where the logfile for the Web Services Producer component should be placed. The default location for UNIX is `$SYBASE/WS-12_5/logs/producer.log`, or `%SYBASE%\WS-12_5\logs\producer.log` for Windows.

`com.sybase.ase.ws.producer.ssl.httpsport`

Indicates the port on which the Web Services Producer should listen for an HTTPS connection. The default entry is 8182.

`com.sybase.ase.ws.producer.ssl.keypassword`

Indicates the password for the SSL certificate. No default is provided.

`com.sybase.ase.ws.producer.ssl.keystore`

Indicates the location of the keystore for SSL. The default location for UNIX is `$SYBASE/WS-12_5/producer/keystore`, or `%SYBASE%\WS-12_5\producer\keystore` for Windows.

`com.sybase.ase.ws.producer.ssl.password`

Indicates the keystore password for SSL. No default is provided.

`com.sybase.ase.ws.producer.tuning.maxidletime`

Indicates the maximum time in milliseconds a thread may remain idle. The default entry is 60000.

`com.sybase.ase.ws.producer.tuning.maxthreads`

Indicates the maximum number of threads in the thread pool servicing the HTTP port. The default entry is 250.

`com.sybase.ase.ws.producer.tuning.minthreads`

Indicates the minimum number of threads in the thread pool servicing the HTTP port. The default entry is 45.

`com.sybase.ase.ws.producer.tuning.ssl.maxidletime`

Indicates the maximum time in milliseconds a thread may remain idle. The default entry is 60000.

`com.sybase.ase.ws.producer.tuning.ssl.maxthreads`

Indicates the maximum number of threads in the thread pool servicing the HTTPS port. The default entry is 250.

`com.sybase.ase.ws.producer.tuning.ssl.minthreads`

Indicates the minimum number of threads in the thread pool servicing the HTTPS port. The default entry is 45.

`com.sybase.ase.ws.ui.activate`

Determines whether the Web-based user interface is activated. The user interface is available at `https://hostname:https_port`. The default entry is true.

myres.properties

The *myres.properties* file is created when the Sybase Central plug-in performs configuration tasks for Adaptive Server Enterprise Web Services. The *myres.properties* file contains the following configuration settings for Adaptive Server Enterprise Web Services.

Table B-2: myres.properties entries

Web Services – Set the following entries to configure Web Services.

`ws.ini`

Indicates the location of the *interfaces* or *sql.ini* file for Adaptive Server Enterprise. The default location for UNIX is `$$SYBASE/interfaces`, or `%SYBASE%\ini\sql.ini` for Windows.

`ws.libtcl`

Indicates the location of the *libtcl.cfg* file used to identify LDAP servers. The default location for 32-bit platforms for UNIX is `$$SYBASE/config/libtcl.cfg`, or `%SYBASE%\ini\libtcl.cfg` for Windows. The default location for 64-bit platforms for UNIX is `$$SYBASE/config/libtcl64.cfg`, or `%SYBASE%\ini\libtcl64.cfg` for Windows.

Producer – Set the following entries to configure the Web Services Producer.

`ws.producer.port`

Indicates the port for the Web Services Producer.

`ws.producer.log`

Indicates the location of the *producer.log* file.

`ws.producer.jettylogfile`

Indicates where the logfile for HTTP requests should be placed. The default location is for UNIX is `$$SYBASE/WS-12_5/logs/http.log`, or `%SYBASE%\WS-12_5\logs\http.log` for Windows.

SSL – Set the following entries to configure SSL.

`ws.ssl.host`

Indicates the name of the SSL host to be accessed.

`ws.ssl.keystorelocation`

Indicates the location of the keystore for SSL. The default location for UNIX is `$$SYBASE/WS-12_5/producer/keystore`, or `%SYBASE%\WS-12_5\producer\keystore` for Windows.

`ws.ssl.certificatepassword`

Indicates the password for the SSL certificate. No default is provided.

`ws.ssl.keystorepassword`

Indicates the keystore password for SSL. No default is provided.

Consumer – Set the following entries to configure the Web Services Consumer and add an entry to the *interfaces* or *sql.ini* file.

`ws.consumer.name`

Indicates the name of the Web Services Consumer as specified in the *interfaces* or *sql.ini* file.

`ws.consumer.host`

Indicates the host machine of the Web Services Consumer.

`ws.consumer.port`

Indicates the port number for the Web Services Consumer process.

`ws.consumer.log`

Indicates the location of the *consumer.log* file.

installws – Set the following entries to run the *installws* script:

`ws.sqlsrv.server_name`

Indicates the name of the Adaptive Server Enterprise on which to run *installws*.

`ws.sqlsrv.sa_login`

Indicates the user login for the Adaptive Server Enterprise.

`ws.sqlsrv.sa_password`

Indicates the password for the Adaptive Server Enterprise.

SOAP and Adaptive Server Enterprise Datatype Mapping

This appendix documents SOAP and Adaptive Server Enterprise datatypes.

Datatype mapping

The following table shows SOAP datatypes and their corresponding types in Adaptive Server Enterprise.

Table C-1: SOAP and Adaptive Server Enterprise datatypes

SOAP datatype	Adaptive Server Enterprise datatype
string	varchar Length depends on Adaptive Server Enterprise page size
boolean	smallint
float	real
double	double precision
decimal	float
duration	datetime
dateTime	datetime
time	datetime
date	datetime
gYearMonth	datetime
gYear	datetime
gMonthDay	datetime
gDay	datetime
gMonth	datetime
hexBinary	Unsupported
base64Binary	Unsupported
anyURI	varchar Length depends on Adaptive Server Enterprise page size
QName	varchar Length depends on Adaptive Server Enterprise page size
NOTATION	varchar Length depends on Adaptive Server Enterprise page size
normalizedString	varchar Length depends on Adaptive Server Enterprise page size
token	varchar Length depends on Adaptive Server Enterprise page size
language	varchar Length depends on Adaptive Server Enterprise page size

SOAP datatype	Adaptive Server Enterprise datatype
NMTOKEN	varchar Length depends on Adaptive Server Enterprise page size
Name	varchar Length depends on Adaptive Server Enterprise page size
NCName	varchar Length depends on Adaptive Server Enterprise page size
ID	varchar Length depends on Adaptive Server Enterprise page size
IDREF	varchar Length depends on Adaptive Server Enterprise page size
ENTITY	varchar Length depends on Adaptive Server Enterprise page size
integer	integer
nonPositiveInteger	integer
negativeInteger	integer
long	integer
int	integer
short	smallint
byte	tinyint
nonNegativeInteger	integer
unsignedLong	integer
unsignedInt	integer
unsignedShort	smallint
unsignedByte	tinyint
positiveInteger	integer

The following table lists datatype conversions between Java, SOAP, and Adaptive Server Enterprise using the Apache Axis toolkit.

Table C-2: Java, SOAP, and ASE datatypes

Java datatype	SOAP datatype	ASE datatype
byte	xsd:base64binary	image
boolean	xsd:boolean	int

Java datatype	SOAP datatype	ASE datatype
byte	xsd:byte	Char (1)
java.util.Calendar	xsd:dateTime	datetime
java.math.BigDecimal	xsd:decimal	Numeric
double	xsd:double	float(8)
float	xsd:float	float
int	xsd:int	int
java.math.BigInteger	xsd:integer	numeric(38,0)
long	xsd:long	float (8)
javax.xml.namespace.Qname	xsd:Qname	varchar(pagesize)
short	xsd:short	short
java.lang.String	xsd:string	varchar(pagesize)

Glossary

This glossary uses the following references:

- *Contrast with* refers to a term that has an opposite or different meaning.
- *Compare with* refers to a term that has a similar meaning.
- *See also* refers to terms that have a related meaning.

Words containing these references are highlighted.

Consumer

The component of Adaptive Server Enterprise Web Services that maps Web methods to proxy tables, allowing a client application to invoke the Web method through a SQL select statement.

document/literal

A type of Web method for which communicating parties specify the data being transmitted and formatted according to XML schemas incorporated into the WSDL file.

DTD

Document Type Definition. A DTD is used to define the legal building blocks of an XML document. A DTD can be declared within an XML document or referenced externally.

LDAP

Lightweight Directory Access Protocol.

Producer

The component of Adaptive Server Enterprise Web Services that enables a client application to access Adaptive Server Enterprise stored procedures and SQL as Web methods.

RPC/encoded

A type of Web method invoked with SOAP messages containing an XML element for each method parameter.

schema

An outline defining the structure, content, and semantics of an XML document.

SDS

Specialty Data Store.

SOAP

Simple Object Access Protocol.

SQL-XML

An XML representation of SQL data.

XML	Extensible Markup Language, a markup language standardized by W3C.
UDDI	Universal Description Discovery and Integration.
URI	A Uniform Resource Identifier. A URI is a string of characters that identify an Internet Resource. The most common URI is the Uniform Resource Locator (URL), which identifies an Internet address. A less common URI is the Universal Resource Name (URN).
WSDL	Web Services Description Language.
XML	Extensible Markup Language.
XPath	XML Path Language.
XQL	XML Query Language. XQL was a precursor of XQuery.
Web method	A function described by WSDL and invoked through SOAP message.
Web service	One or more Web methods described by a WSDL file.
Xerces	The Apache open-source XML parser.

Index

Symbols

- () (parentheses) xii
- , (comma) xii
- [] (square brackets) xii
- { } (curly braces) xii

A

- Adaptive Server Enterprise Web Services
 - advantages 4
 - configuration 23
 - Consumer 17
 - directory tree 81
 - licensing 25
 - log files 55
 - logging 54
 - methods 38
 - Producer 15
 - security 26
 - using 33
- Apache
 - sample client 59

B

- Backus Naur Form (BNF) notation xii
- bin directory 81
- brackets. *See* square brackets []

C

- case sensitivity
 - in SQL xiii
- certificate for Microsoft .NET 28
- certifications xi
- CodeXchange xi

- comma (,) xii
- configssl 27
- configuration 23
 - files 26
 - logging.properties 26
 - logging.properties file 26
 - properties 87
 - ws.properties 26
 - ws.properties file 26
 - wsmmsg.properties file 26
- configuring
 - after installation 24
 - during installation 24
- Consumer 17
- consumer.log 55
- conventions xii
 - See also* syntax
- curly braces ({}) xii

D

- datatypes
 - SOAP and Adaptive Server Enterprise 91
- diagnostic tools 75
- directories
 - bin 81
 - lib 83
 - logs 83
 - producer 84
 - props 84
 - samples 84

E

- EBFs and software updates xii
- execute failure 72
- execute method 38
 - examples 40

Index

- parameters 38
- syntax 38

Extensible Markup Language 6

G

Getting Started CD x

H

handlers

- HTTP 16
- SOAP 16

HTTP handler 16

http.log 55

I

installation

- contents 81

J

JDBC-level tracing 75

K

known issues 69

- abnormal termination of sp_webservices 73
- command-line arguments 71
- entries in ws.properties 71
- locating WSDL 71
- null passwords 73
- runconsumer or stopconsumer fails 72
- runexecute or execute fails 72
- runproducer or stopproducer fails 72
- runws or stopws fails 72
- specifying SOAP endpoints with SSL 73
- starting the Producer and Consumer 70
- Web Services proxy table restrictions 74

L

lib directory 83

licensing 25

log files 55

log4j 55

logging 54

- consumer.log 55
- http.log 55
- producer.log 55
- setting policies 55

logging.properties file 26

login method 40

- parameters 40, 46, 47, 48
- syntax 40, 46, 47, 48
- usage 41

logout method 41

- syntax 41
- usage 41

logs directory 83

M

messages 75

methods 38

- execute 38
- login 40
- logout 41

Microsoft .NET

- installing SSL certificate 28
- sample client 64

myres.properties file

- contents 89

N

null passwords 73

P

Producer 15

- HTTP handler 16
- SOAP handler 16
- starting 33, 42

- stopping 36, 44
- using 33, 42
- XML mapper 16
- producer directory 84
- producer.log 55
- Product Manuals Web site x
- properties
 - com.sybase.ase.ws.consumer.cisport 87
 - com.sybase.ase.ws.consumer.logfilename 87
 - com.sybase.ase.ws.interfaces 87
 - com.sybase.ase.ws.libtcl 87
 - com.sybase.ase.ws.producer.httpport 87
 - com.sybase.ase.ws.producer.jettylogfile 88
 - com.sybase.ase.ws.producer.logfilename 88
 - com.sybase.ase.ws.producer.ssl.httpsport 88
 - com.sybase.ase.ws.producer.ssl.keypassword 88
 - com.sybase.ase.ws.producer.ssl.keystore 88
 - com.sybase.ase.ws.producer.ssl.password 88
 - com.sybase.ase.ws.producer.tuning.maxidletime 88
 - com.sybase.ase.ws.producer.tuning.maxthreads 88
 - com.sybase.ase.ws.producer.tuning.minthreads 88
 - com.sybase.ase.ws.producer.tuning.ssl.maxidletime 88
 - com.sybase.ase.ws.producer.tuning.ssl.maxthreads 88
 - com.sybase.ase.ws.producer.tuning.ssl.minthreads 88
 - com.sybase.ase.ws.ui.activate 88
 - ws.consumer.host 90
 - ws.consumer.log 90
 - ws.consumer.name 90
 - ws.consumer.port 90
 - ws.ini 89
 - ws.libtcl 89
 - ws.producer.jettylogfile 89
 - ws.producer.log 89
 - ws.producer.port 89
 - ws.sqlsrv.sa_login 90
 - ws.sqlsrv.sa_password 90
 - ws.sqlsrv.server_name 90
 - ws.ssl.certificatepassword 89
 - ws.ssl.host 89
 - ws.ssl.keystorelocation 89

- ws.ssl.keystorepassword 89
- props directory 84

R

- runconsumer failure 72
- runexecute failure 72
- runproducer failure 72
- runws failure 72

S

- sample applications 59
 - Apache client 59
 - Execute.exe 64
 - Microsoft .NET client 64
 - runexecute 60
- samples directory 84
- security 26
- Simple Object Access Protocol, *See* SOAP 10
- SOAP 10
 - message structure 11
- SOAP handler 16
- SOAP stack 18
- SQL case sensitivity xiii
- square brackets [] xii
- SSL
 - configuring 27
- stopconsumer failure 72
- stopproducer failure 72
- stopws failure 72
- Sybase certifications on the Web xi
- symbols xii
- syntax conventions xii

T

- Technical Library CD x
- Technical Support xiv
- troubleshooting 69

Index

U

Using Adaptive Server Enterprise Web Services 33

W

Web services

 overview 3

Web Services Description Language, *See* WSDL 9

ws.properties

 entries 71

ws.properties file 26

 contents 87

WSDL 9

 locating 71

wsmg.properties file 26

X

XML 6

 document structure 7

XML mapper 16