Web Services User’s Guide

Adaptive Server® Enterprise

12.5.1

DOCUMENT ID: DC10061-01-1251-01

LAST REVISED: September 2003

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

Contents

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”
- Part 2, “Configuration”
    - Chapter 2, “Configuring Adaptive Server Enterprise Web Services”
- Part 3, “Administration”
    - Chapter 3, “Using Adaptive Server Enterprise Web Services”
    - Chapter 4, “Sample Applications”
- Part 4, “Troubleshooting”
    - Chapter 5, “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* – the basics for understanding and investigating performance questions in Adaptive Server.
  - *Locking* – describes how the various locking schemas can be used for improving 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 Sybase's Failover 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.

v 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.

v 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

v 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.	*System Administration Guide* *sql.ini* file *column_name* *$SYBASE/ASE* directory

Variables—or 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` *`column_name`*
	`from` *`table_name`*
	`where` *`search_conditions`*

Type parentheses as part of the command.	`compute` *row_aggregate* `(`*column_name*`)`
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.	`{cash, check, credit}`
Brackets mean that to choose one or more of the enclosed options is optional. Do not type the brackets.	`[cash | check | credit]`
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.	`cash, check, credit`
The pipe or vertical bar(|) means you may select only one of the options shown.	`cash | check | credit`

<table>
<tr><th>Element</th><th>Example</th></tr>
<tr><td>An ellipsis (...) means that you can <em>repeat</em> the last unit as many times as you like.</td><td><code>buy thing = price [cash | check | credit]</code>
<code>[, thing = price [cash | check | credit]]...</code>
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.</td></tr>
</table>

- 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, please 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.

CHAPTER 1

Understanding Adaptive Server Enterprise Web Services

This chapter discusses the following:

Overview

A Web service is a self-contained, modular application that can be accessed and used over a network connection using the Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), 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.

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.

- The Web Services Producer component enables client applications to access SQL and stored procedures in Adaptive Server Enterprise using the Simple Object Access Protocol (SOAP).
- 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.

Note The Web Services Consumer component of Adaptive Server Enterprise Web Services is not available in this release. For the most current information about Adaptive Server Enterprise Web Services, go to Product Manuals at http://www.sybase.com/support/manuals/.

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 are returned according to the SQLX standard, and the client application can receive XML data, schema, and DTDs.

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 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.

Security

The Web Services Producer 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).

Web services standards

Web services are structured with XML, described with WSDL, and transferred with SOAP. Adaptive Server Enterprise Web Services enables client applications to access Web services using LDAP.

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 mutually 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.

The following is a DTD for the previous example for discount information:

```
<!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="sqlxOptions" 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 sqlxOptions 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.

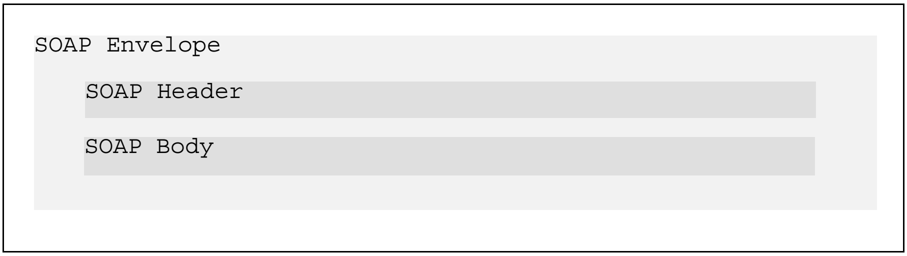
SOAP

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 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



The 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 is a SOAP response message corresponding 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/

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.

Adaptive Server Enterprise Web Services components

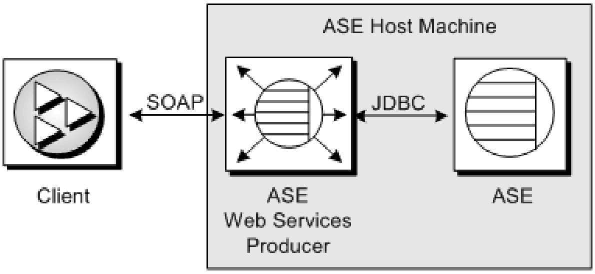
Adaptive Server Enterprise Web Services consists of two components that run independently of Adaptive Server Enterprise:

- Web Services Producer
- Web Services Consumer

Web Services Producer

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.

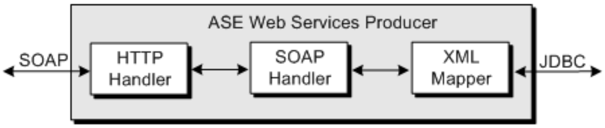
Figure 1-4: Web Services Producer



The client can send a SQL or stored procedure command as a SOAP request, and any result is returned as a SOAP response. The methods provided by the Web Services Producer component conform to the SQLX standard.

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

Figure 1-5: 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.

Web Services Consumer

The Web Services Consumer component of Adaptive Server Enterprise Web Services is not available in this release.

PART 2

Configuration

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

CHAPTER 2

Configuring Adaptive Server Enterprise Web Services

Adaptive Server Enterprise Web Services is installed as part of the installation for Adaptive Server Enterprise. For installation instructions, 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	17
Configuration files	19
Security	20

Licensing

Although it is installed with Adaptive Server Enterprise, Adaptive Server Enterprise Web Services is a separately licensed feature, and it requires a license registered through SySAM named ASE_WEBSERVICES. Before configuring Adaptive Server Enterprise Web Services, make sure that your license is registered through SySAM. For more information on SySAM, see the *Installation Guide* for Adaptive Server Enterprise.

Configuration

Use the following procedure to configure Adaptive Server Enterprise Web Services.

v Configuring Adaptive Server Enterprise Web Services

1 Provide values for the following properties in the *ws.properties* file in the *props* directory, or use the provided defaults:

- `com.sybase.ase.ws.interfaces`
- `com.sybase.ase.ws.producer.httpport`
- `com.sybase.ase.ws.producer.jettylogfile`
- `com.sybase.ase.ws.producer.logfilename`
- `com.sybase.ase.ws.producer.tuning.maxidletime`
- `com.sybase.ase.ws.producer.tuning.maxthreads`
- `com.sybase.ase.ws.producer.tuning.minthreads`
- `com.sybase.ase.ws.producer.tuning.ssl.maxidletime`
- `com.sybase.ase.ws.producer.tuning.ssl.maxthreads`
- `com.sybase.ase.ws.producer.tuning.ssl.minthreads`

2 Add your certificates for using SSL, or allow a non-trusted certificate to be automatically generated.

- To add an SSL certificate, run the *configssl* script, which can be found in the *bin* directory:

  ```
  configssl -h <httpsPort> -c
  <certificate_password> -s <keystore_password>
  ```

  The *configssl* script sets the following entries in the *ws.properties* file:

  - `com.sybase.ase.ws.producer.ssl.httsport`
  - `com.sybase.ase.ws.producer.ssl.keypassword`
  - `com.sybase.ase.ws.producer.ssl.keystore`
  - `com.sybase.ase.ws.producer.ssl.password`

  For details on the parameter values for the *configssl* script, see "Configuring SSL" on page 20.

- To replace a certificate that was auto-generated in the keystore with your own certificate, use the java keytool utility. This utility is in the *bin* directory for your JRE. For more information on the keytool utility, see the documentation at the following URL:

  http://java.sun.com/j2se/1.3/docs

  If you are using Microsoft .NET, you must install a certificate. For details on installing a certificate for Microsoft .NET, see "Installing a certificate for Microsoft .NET" on page 21.

3 Verify that the *installjconnect* script has been run. This script installs jConnect and can be found in the *scripts* directory for your Adaptive Server Enterprise installation.

4 Verify that the necessary sp_configure options are set on your Adaptive Server Enterprise by executing the following stored procedure command in isql:

```
sp_configure ‘enable webservices’, 1
```

5 Provide configuration parameter values for remaining entries in the *ws.properties* and *logging.properties* files in the *props* directory.

6 Verify that you have JRE version 1.3 or later installed. This version should be installed with the latest version of Adaptive Server Enterprise.

Configuration files

The *props* directory contains the following configuration files:

- ws.properties
- logging.properties

ws.properties

The *ws.properties* file contains configuration settings for Adaptive Server Enterprise Web Services. For details on these configuration properties, see the Appendix, “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 34.

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*.

Security

To ensure secure operation of Adaptive Server Enterprise Web Services, Sybase recommends 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

To 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>
```

Note You can also add your own certificate for SSL.

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.

Installing a certificate for Microsoft .NET

Use the following procedure to install a certificate for Microsoft .NET.

v 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 26.

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 the View Certificate button. The Certificate dialog box appears.

4 Click the Install Certificate button. The Certificate Manager Import Wizard opens.

5 Click the Next button 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.”

v 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.

CHAPTER 3 Using Adaptive Server Enterprise Web Services

This chapter covers the following topics:

Note The Web Services Consumer component of Adaptive Server Enterprise Web Services is not available in this release. For the most current information about Adaptive Server Enterprise Web Services, go to Product Manuals at http://www.sybase.com/support/manuals/.

Before using Adaptive Server Enterprise Web Services, make sure you have completed the configuration tasks in Chapter 2, “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 can be found in the *bin* directory:

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

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.

Restrictions

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

- 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*.

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

v 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 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.

v Verifying that the Web Services Producer component is running

To verify that the Web Services Producer component is running:

- Enter the following URL in the location window of your browser:

  ```
  http://producer_host:producer_port
  ```

  where:

  - *producer_host* indicates the machine on which the Adaptive Server Enterprise resides.
  - *producer_port* indicates the port for the Web Services Producer component.

  If you are using SSL, use a URL like the following:

  ```
  https://producer_host:producer_ssl_port
  ```

  where:

  - *host_machine* indicates the machine on which the Adaptive Server Enterprise resides.
  - *producer_ssl_port* indicates the port for the Web Services Producer component.

  A browser error indicates that the Web Services Producer component is not running.

Stopping the Web Services Producer

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

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

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.

Restrictions

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

- 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*.

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

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

The Web Services Producer component provides three methods:

- execute – used to execute a SQL statement or stored procedure.
- login – used to establish 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 T-SQL statement or stored procedure in Adaptive Server Enterprise.

Syntax

```
execute aseServerName userName password  sqlxOptions
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.

Note Due to a limitation with JDBC, print statements are not captured when stored procedures are executed through Adaptive Server Enterprise Web Services. Other limitations concerning the use of SQL queries with JDBC also apply to Adaptive Server Enterprise Web Services.

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”
```

Note Due to a limitation with JDBC, print statements are not captured when stored procedures are executed through Adaptive Server Enterprise Web Services. Other limitations concerning the use of SQL queries with JDBC also apply to Adaptive Server Enterprise Web Services.

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

The Web Services Consumer component of Adaptive Server Enterprise Web Services is unavailable for this release.

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:

1. Adaptive Server Enterprise Web Services log files
2. 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.

Note The Web Services Consumer component of Adaptive Server Enterprise Web Services is not available in this release. For the most current information about Adaptive Server Enterprise Web Services, go to Product Manuals at http://www.sybase.com/support/manuals/.

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

By default, there is no rollover policy for logging in Adaptive Server Enterprise Web Services. You can create a log rollover policy based on time or on file size.

Time-based policies

You can configure a time-based log policy to roll over at the following intervals:

- The start of each month.
- The start of each week. This interval depends on locale.
- The start of each day. Each day begins at midnight.
- The start of each half day. This means every midnight and noon.
- The start of each hour.
- The start of each minute.

For example, to create a log policy that rolls over the log files every night at midnight, edit the following line in the *logging.properties* file:

```
log4j.appender.P=org.apache.log4j.FileAppender
```

The new entry should read as follows (on one line):

```
log4j.appender.FILE=
org.apache.log4j.DailyRollingFileAppender
```

Size-based policies

You can configure a log policy based on the size of the logging files. For example, to create a log policy that rolls over a log file at 10MB and keeps one backup of the log file, edit the following line in the *logging.properties* file:

```
log4j.appender.P=org.apache.log4j.FileAppender
```

The new entry should read as follows (on one line):

```
log4j.appender.P=
org.apache.log4j.RollingFileAppender
```

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

CHAPTER 4 Sample Applications

Tools are provided under the *samples* directory to create and run sample clients for Apache Axis and Microsoft .NET.

This chapter covers the following topics:

Topic	**Page**
Web Services Producer component sample applications	37
Web Services Consumer component sample applications	45

Web Services Producer component sample applications

The following sample applications are provided for the Web Services Producer component:

- Apache sample client
- Microsoft .NET sample client

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.

v **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 T-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 - return an XML schema
  - dtd - return an XML DTD
  - data - return a result set
  - all - return 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.

Example1

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:

```
<!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/
sqlx">
 <xsd:import
  namespace="http://www.w3.org/2001/XMLSchema"
  schemaLocation="http://www.iso-standards.org/mra/
9075/sqlx.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.

v **Creating the sample client**

- Make sure the JRE variable points to your JRE. You must use JRE version 1.3.1 or later. By default, the JRE supplied in the UNIX *$SYBASE_JRE* directory or the Windows *%SYBASE_JRE%* directory is used.

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 T-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 - return an XML schema
  - dtd - return an XML DTD
  - data - return a result set
  - all - return 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.

Example1

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.

Web Services Consumer component sample applications

The Web Services Consumer component of Adaptive Server Enterprise Web Services is unavailable for this release. There are currently no sample applications provided for the Web Services Consumer component.

PART 4

Troubleshooting

This section describes troubleshooting procedures and error messages.

CHAPTER 5 Troubleshooting

This chapter documents known issues and corresponding user actions.

Known issues

The following are known issues that can occur with Adaptive Server Enterprise Web Services. If you encounter these problems, follow the user actions described.

Starting the Producer

Issue

The *runproducer* script does not successfully start the Web Services Producer component.

User action

1 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:

```
java -version
```

2 If you want the Web Services Producer component to run with a properties file other than *ws.properties*, you must specify the absolute path for the file. For example,

```
C:\sybase\WS-12_5\bin\runproducer -f
C:\sybase\WS-12_5\props\myfile.properties
```

Issue

The Web Services Producer component finds the specified *ase_service_name* in the *interfaces* file, but the *producer.log* shows the following error messages:

```
ERROR [main] - Error locating libtcl.cfg file.
ERROR [main] - java.io.FileNotFoundException: LDAP
config File does not exist
```

User action

No user action is required.

Executing a Web method

Issue

An Adaptive Server Enterprise Web Services method fails to execute, and the stack trace reports the method is not found.

User action

Make sure you have the correct JRE installed. Adaptive Server Enterprise Web Services requires JRE 1.3.1 or later.

Check the *producer.log* file to see which version of JRE is being used.

Locating WSDL

Issue

A client connecting to the Web Services Producer through a Web browser cannot find the WSDL file.

User action

1. Verify that the Web Services Producer is running.
2. Make sure the browser URL indicates `https://` for an SSL connection and `http://` for a standard connection.

Java Virtual Machine crashes

Issue

A bug in the Java Native Interface (JNI) can cause the Java Virtual Machine (JVM) to crash, after an application has terminated, with an error message similar to the following:

```
Another exception has been detected while
we were handling last error.Dumping information about
last error.

ERROR REPORT FILE = (N/A)
PC                = 0x0x41621c7a
SIGNAL            = 11
FUNCTION NAME     = (N/A)
LIBRARY NAME      = (N/A)
Please check ERROR REPORT FILE for
further information, if there is any.
Good bye.
```

This problem occurs on Red Hat SMP Linux 7.1. This problem may occur in client applications that use JNI and invoke Adaptive Server Enterprise Web Services methods. The problem may also occur on the Microsoft Windows and Solaris platforms.

User action

Add an explicit exit call at the end of the main method for the application. For example:

```
Runtime.getRuntime().exit(0)
```

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 instead. For example:

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

Command line arguments

Issue

Scripts do not run when no space is placed between arguments and argument values. For example, the following invocation of the *configssl* script will not execute the script:

```
configssl -dhostname
```

User action

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

```
configssl -d hostname
```

runproducer or *stopproducer* fails

Issue

The *runproducer* or *stopproducer* script fails to execute.

User action

If 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* 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*, *stopproducer*, and *runexecute* 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.

PART 5

Appendixes

This section contains the appendixes for this book.

APPENDIX A 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
bin	Scripts for configuring and running Adaptive Server Enterprise Web Services components.
lib	Java libraries and packages used by Adaptive Server Enterprise Web Services.
logs	Default location for log files.
producer	Files and subdirectories for Web Servives Producer component.
props	Files for Adaptive Server Enterprise Web Services properties.
samples	Sample scripts for building and running a sample client with the Web Services Producer component.

bin **directory contents**

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

Table A-2: bin directory contents

File/Directory name	Function
configssl	Configures SSL.
runconsumer	Starts Adaptive Server Enterprise Web Services Consumer. **Note** The Web Services Consumer component of Adaptive Server Enterprise Web Services is not available in this release. For the most current information about Adaptive Server Enterprise Web Services, go to Product Manuals at http://www.sybase.com/support/manuals/.
runproducer	Starts Web Services Producer.
runtcpmon	Creates a monitor to trace SOAP messages.

lib directory contents

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

Table A-3: lib directory contents

File/Directory name	Function
axis.jar	Apache Axis file.
commons-discovery.jar	Apache Axis file.
commons-logging.jar	Apache Axis file.
flexlm.jar	Library for FLEXlm licensing.
jTDS2.jar	JDBC library.
jTDS2d.jar	JDBC library.
javax.servlet.jar	Servlet library.
jaxrpc.jar	Apache Axis file.
jcert.jar	JSSE library.
jconn2.jar	JDBC library.
jconn2d.jar	JDBC library.
jnet.jar	JSSE library.
jsse.jar	JSSE library.
log4j-1.2.4.jar	log4j logger.
mail.jar	This information is currently unavailable.
org.mortbay.jetty-jdk1.2.jar	HTTP server.
saaj.jar	Apache Axis file.
sqlx.jar	SQLX file.
tools.jar	Compiler.
ws-12.5.jar	Code for Web Services Consumer, Web Services Producer.
ws-12.5d.jar	Debugger Code for Web Services Consumer, Web Services Producer.
wsdl4j.jar	Apache Axis file.
xercesImpl.jar	Xerces parser.
xmlParserAPIs.jar	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
consumer.log	Logs Web Services Consumer component activity. **Note** The Web Services Consumer component of Adaptive Server Enterprise Web Services is not available in this release. For the most current information about Adaptive Server Enterprise Web Services, go to Product Manuals at http://www.sybase.com/support/manuals/.
http.log	Logs Web server activity.
producer.log	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
keystore	Holds encryption keys.
server	Directory structure required for Jetty.
WEB-INF	Directory structure required for Jetty.
wscertificate.cer	Auto-generated certificate for SSL. This file is present only the *configssl* 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
logging.properties	Configuration file for log4j.
ws.properties	All configuration parameters for Web Services Consumer, Web Services Producer.
wsmsg.properties	Configuration file for messages.

samples directory contents

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

Table A-7: samples directory contents

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

APPENDIX B Configuration Properties

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

Adaptive Server Enterprise Web Services configuration properties

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

Table B-1: ws.properties entries

`com.sybase.ase.ws.interfaces` Indicates the location of the *interfaces* or *sql.ini* file for Adaptive Server Enterprise. The default location for UNIX is *$SYBASE/interfaces*, or *%SYBASE%\ini\sql.ini* for Windows.
`com.sybase.ase.ws.libtcl` Indicates the location of the *libtcl.cfg* file used to identify LDAP servers. The default location for 32-bit platforms for UNIX is *$SYBASE/config/libtcl.cfg*, or *%SYBASE%\ini\libtcl.cfg* for Windows. The default location for 64-bit platforms for UNIX is *$SYBASE/config/libtcl64.cfg*, or *%SYBASE%\ini\libtcl64.cfg* for Windows.
`com.sybase.ase.ws.producer.httpport` Indicates the port on which the Web Services Producer should listen for an HTTP connection. The default entry is “8181.”
`com.sybase.ase.ws.producer.jettylogfile` Indicates where the logfile for HTTP requests should be placed. The default location is for UNIX is *$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.httsport`

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.”

APPENDIX C

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

SOAP datatype	Adaptive Server Enterprise datatype
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.
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

SOAP datatype	Adaptive Server Enterprise datatype
int	integer
short	smallint
byte	tinyint
nonNegativeInteger	integer
unsignedLong	integer
unsignedInt	integer
unsignedShort	smallint
unsignedByte	tinyint
positiveInteger	integer

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.

attribute
A property of an information object.

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.

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

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 schema
Extensible Markup Language.

XPath
XML Path Language.

XQL
XML Query Language. XQL was a precursor of XQuery.

Index