



Web 服务用户指南

Adaptive Server[®] Enterprise

15.7

文档 ID: DC20143-01-1570-01

最后修订日期: 2011 年 9 月

版权所有 © 2011 Sybase, Inc. 保留所有权利。

本出版物适用于 Sybase 软件及所有后续版本, 除非在新版本或技术说明中另有说明。此文档中的信息如有更改, 恕不另行通知。此处说明的软件按许可协议提供, 其使用和复制必须符合该协议的条款。

若要订购附加文档, 美国和加拿大的客户请拨打客户服务部门电话 (800) 685-8225 或发传真至 (617) 229-9845。

持有美国许可协议的其它国家 / 地区的客户可通过上述传真号码与客户服务部门联系。所有其他国际客户请与 Sybase 子公司或当地分销商联系。仅在定期安排的软件发布日期提供升级。未经 Sybase, Inc. 的事先书面许可, 本书的任何部分不得以任何形式、任何手段 (电子的、机械的、手动、光学的或其它手段) 进行复制、传播或翻译。

Sybase 商标可在 <http://www.sybase.com/detail?id=1011207> 处进行查看。Sybase 和文中列出的标记均是 Sybase, Inc. 的商标。® 表示已在美国注册。

SAP 和此处提及的其它 SAP 产品与服务及其各自的徽标是 SAP AG 在德国和世界各地其它几个国家 / 地区的商标或注册商标。

Java 和所有基于 Java 的标记都是 Sun Microsystems, Inc. 在美国和其它国家 / 地区的商标或注册商标。

Unicode 和 Unicode 徽标是 Unicode, Inc. 的注册商标。

提到的所有其它公司和产品名均可能是与之相关的相应公司的商标。

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。

目录

第 1 部分	简介	
第 1 章	了解 Adaptive Server Enterprise Web 服务	3
	概述	3
	Adaptive Server Enterprise Web 服务	4
	ASE Web 服务的优点	4
	存储过程和函数	4
	SQL	4
	安全性	5
	LDAP	5
	用户定义的 Web 服务	5
	Web 服务标准	5
	XML	6
	WSDL	9
	SOAP	10
第 2 章	了解 ASE Web 服务引擎	13
	Web 服务的生产者	13
	生产者组件	14
	生产者 Web 方法	15
	用户定义的 Web 服务	15
	Web 服务的使用者	16
	使用者组件	17
	代理表	18

第 2 部分

配置

第 3 章

配置 ASE Web 服务	21
配置	21
在安装期间进行配置	22
在安装完成后进行配置	22
许可	23
配置文件	23
ws.properties 文件	23
logging.properties 文件	23
wsmsg.properties 文件	24
安全性	24
配置 SSL	24
为 Microsoft .NET 安装证书	26

第 3 部分

管理

第 4 章

使用 ASE Web 服务	29
使用 ASE Web 服务引擎	29
启动和停止 ASE Web 服务引擎	29
ASE Web 服务方法	31
使用 sp_webservices	35
调用 Web 服务	38
使用用户定义的 Web 服务	40
用户定义的 Web 服务的命令	41
将 sp_webservices 与用户定义的 Web 服务一起使用	45
用户定义的 Web 服务的安全性	47
用户定义的 Web 服务的审计	48
ASE Web 服务日志记录	49
ASE Web 服务日志文件	49
转滚日志文件	50
使用 Sybase Central	50

第 5 章

示例应用程序	51
Apache 示例客户端	51
创建示例客户端	51
使用 runexecute	52
Microsoft .NET 示例客户端	55
创建示例客户端	56
使用 Execute.exe	56

第 4 部分

故障排除

第 6 章

故障排除	61
故障排除问题	61
远程服务器类定义设置	61
未映射 RPC/ 编码 Web 方法	62
文档 / 文字结果被截断	62
启动 ASE Web 服务引擎	63
定位 WSDL	63
指定 ws.properties 中的条目	63
Windows NT 命令行参数	64
运行或停止脚本失败	64
空口令	64
指定使用 SSL 的 SOAP 结束点	65
sp_webservices 'add' 异常终止	65
Web 服务代理表限制	65
sysattributes 表条目	66
诊断工具	66
详细记录	67
启用 JDBC 级跟踪	67
消息	68

第 5 部分

附录

附录 A

安装内容	73
ASE Web 服务目录树	73
bin 目录内容	74
lib 目录内容	74
logs 目录内容	75
producer 目录内容	75
props 目录内容	75
samples 目录内容	76

附录 B

配置属性	77
ws.properties	77
myres.properties	79
指定属性文件条目	80

附录 C	SOAP 和 Adaptive Server Enterprise 的数据类型映射	81
	数据类型映射	81
	SOAP 到 ASE 数据类型映射	81
	create service 命令的 ASE 到 SOAP 数据类型映射	84
词汇表		85
索引		87

简介

本部分介绍 Adaptive Server[®] Enterprise Web 服务。



了解 Adaptive Server Enterprise Web 服务

本章讨论以下内容：

主题	页码
概述	3
Adaptive Server Enterprise Web 服务	4
ASE Web 服务的优点	4
Web 服务标准	5

概述

Web 服务是自我包含的、模块化的应用程序，可以通过网络连接进行访问。最终用户通过使用 Web 服务，可以牺牲部分性能来换取更高的互用性。互用性是通过遵守简单对象访问协议 (SOAP)、Web 服务描述语言 (WSDL) 以及可扩展标记语言 (XML) 开放式标准来强制执行的。

不管 Web 服务是使用哪种编程语言实现的，都可以从许多不同的平台和操作系统进行访问，从而大大提高了不同应用程序共享数据的能力。通过使用许多互不关联的 Web 服务（每个 Web 服务处理一组有限的特定任务），企业可以在一个安全和受控的环境中公开自己现有的软件，从而动态地集成软件并不断地扩展。由于提供了标准化的方法来调用远程应用程序，Web 服务减少了基础结构所需的代码量。通过允许用户从公开的接口 (WSDL) 提取实现，Web 服务提供了构建面向服务的体系结构 (SOA) 所需的工具。

Adaptive Server Enterprise Web 服务

Adaptive Server Enterprise (ASE) Web 服务包括独立于 Adaptive Server Enterprise 运行的 ASE Web 服务引擎。

ASE Web 服务引擎提供了以下功能：

- 允许客户端应用程序使用 SOAP 访问 Adaptive Server Enterprise 中的 SQL 和存储过程。
- 允许 Adaptive Server Enterprise 访问其它应用程序的 Web 服务。这些外部 Web 服务在运行期映射到 Adaptive Server Enterprise 代理表。
- 提供用户定义的 Web 服务，它允许在 Adaptive Server Enterprise 中用 Web 浏览器或 SOAP 客户端执行 SQL 命令。

有关 ASE Web 服务引擎的详细信息，请参见第 2 章 “[了解 ASE Web 服务引擎](#)”

ASE Web 服务的优点

使用 ASE Web 服务引擎，用户可以使用存储过程、用户定义的函数以及 SQL 来查询和处理数据。客户端应用程序可以发送一个包含 SQL 命令的 SOAP 请求，然后通过 SOAP 接收结果。数据将按 SQLX 标准返回，客户端应用程序可以收到 XML 数据、模式和 DTD。

存储过程和函数

存储过程可将数据的内部逻辑视图从业务层逻辑中分离出来，并扩展 SQL 的影响和性能。存储过程还可以通过远程方式执行。用户可以使用存储过程和用户定义的函数来调用 Java 方法（按 ANSI SQLJ 标准的指定），并检索标准 XML 格式的数据。

SQL

因为 SQL 可用于处理 XML 数据，所以支持 SOAP 的客户端应用程序可以使用 ASE Web 服务引擎来管理 Adaptive Server Enterprise 中的数据。通过 ASE Web 服务引擎，SQL 还可以用来调用 Web 服务。

安全性

ASE Web 服务安全性功能包括安全套接字层 (SSL)，并且提供了重要的数据库安全和授权功能，例如通过轻量目录访问协议 (LDAP) 进行访问控制。

LDAP

LDAP 是一种 Internet 协议，用于访问分布式环境中的目录。LDAP 服务器存储着在各资源之间建立连接和授予目录访问权限所需的用户信息，这样客户端应用程序就无需知道这些信息。ASE Web 服务允许客户端应用程序使用 LDAP 访问 Web 方法。

ASE Web 服务支持 LDAP 版本 3 服务器。有关使用 LDAP 启用用户鉴定和定位 Adaptive Server Enterprise 数据服务器的更详细信息，请参见 Adaptive Server Enterprise 的《安全性管理指南》。

用户定义的 Web 服务

使用用户定义的 Web 服务，可以在 Adaptive Server Enterprise 中用 Web 浏览器或 SOAP 客户端执行 SQL 命令。此功能可用于定义 Web 服务的名称、要执行的 SQL 以及 URL 的位置。

使用用户定义的 Web 服务，可以创建一个符合 SOA 标准的、到 Adaptive Server Enterprise 的接口。

Web 服务标准

Web 服务是用 XML 构建的，它采用 WSDL 进行描述，并通过 SOAP over HTTP 传输数据。使用 ASE Web 服务，客户端应用程序可以访问 Web 服务，并且可以使用远程 Web 服务。

XML

XML 用于描述数据。XML 是从 SGML 派生而来的，具有其它标记语言（如 HTML）的某些特性。不过，XML 的标记是由用户定义的，因此可以进行扩展。正是由于这个特点，XML 是在两个或多个通信应用程序可以理解的结构中交换数据的一种理想方法。

示例

下面的对 `pubs2` 数据库进行的 `isql` 查询可查找折扣信息：

```
select * from discounts
```

此查询生成以下结果集：

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

此结果集可以采用多种方式用 XML 表示。下面是 ASE Web 服务生成的一种 XML 表示法，该表示法使用了 SQLX 格式（SQLX 是 ANSI 标准的一部分，专用于 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>

```

第一行描述 XML 的版本和字符编码。其余标记是由用户定义的，用于描述文档的结构和数据。通过这些用户定义的标记，用户可以针对特定的应用程序来自定义文档，例如利用折扣信息计算价格的应用程序。

XML 文档结构

在格式正确的 XML 文档中，用户定义的元素及元素排列是由文档类型定义 (DTD) 或 XML 模式定义的。

下面是上一个折扣信息示例的 DTD:

```

<!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)>
]>

```

下面是上一个折扣信息示例的 XML 模式的一部分:

```

<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:sequence>
  </xsd:complexType>

```

```
        <xsd:element name="discount" type="FLOAT" />
    </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:maxLength value="40"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="VARCHAR_4">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="4"/>
    </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="FLOAT">
    <xsd:restriction base="xsd:float"/>
</xsd:simpleType>
    <xsd:element name="ws" type="TableType.ws"/>
</xsd:schema>
```

XML 模式或 DTD 可以作为其所描述的 XML 文档的一部分包含在其中，或者作为独立的文件由 XML 文档引用。XML 模式和 DTD 的文件后缀名分别为 *.xsd* 和 *.dtd*。

有关 XML 的详细信息，请参考以下文档：

- World Wide Web 协会 (W3C)，网址是 <http://www.w3.org>
- 位于 <http://www.w3.org/XML/> 上的 W3C 可扩展标记语言 (XML)

WSDL

WSDL 文档是用 XML 编写的，用于描述 Web 服务。除了指定 Web 服务的位置以外，WSDL 描述还可以使用以下标记，指定由 Web 服务提供的方法以及 Web 服务所使用的消息、数据类型和通信协议：

- **<service>** 定义 Web 服务的名称。例如，名为 **ExecuteStoredProcService** 的 Web 服务可以按以下方式命名：

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

WSDL 文档可以包含一个或多个 **<service>** 标记。对于 ASE Web 服务引擎，则仅有一个名为“ase”的服务。

- **<binding>** 定义所使用的通信协议。下面的示例使用的是 SOAP 协议：

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

WSDL 还支持使用 HTTP 和 MIME 协议。

- **<port>** 指定 Web 服务地址。例如：

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

<port> 标记具有名称和绑定的属性。

- **<message>** 定义所使用的消息。例如：

```
<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>
```

这是一个用于名为 **executeRequest** 的方法的请求消息。**<part>** 标记与请求消息中方法调用的参数值以及响应中的返回值相对应。

- `<operation>` — 将消息与 Web 方法请求或响应相关联。例如：

```
<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>` — 定义所提供的方法。`<operation>` 标记是 `<portType>` 的子元素。例如：

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

- `<types>` — 定义所使用的数据类型。WSDL 使用 XML 模式语法来定义数据类型。

WSDL 通常是由 ASE Web 服务引擎自动生成的，可以使用 Web 浏览器在以下位置查看：

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

其中 *myserver* 是运行 ASE Web 服务引擎的主机的名称，*producer_port* 是端口号。

SOAP

SOAP 是一种基于 XML 的、与平台和语言无关的协议，用于在应用程序之间发送消息和数据。SOAP 定义消息的结构，描述处理消息的方式，并且提供对应用程序定义的数据类型进行编码的规则。SOAP 允许应用程序使用任意的标准传输层协议（通常是 HTTP）来发送和接收远程过程调用 (RPC)。

SOAP 消息结构

SOAP 消息由标头和正文组成，这两部分都包含在一个 SOAP 信封中。一般而言，SOAP 请求消息不包含标头信息，但与前一请求消息对应的响应消息则包含标头，并且不一定会显示消息正文。

有关 SOAP 的详细信息，请参考以下文档：

- 简单对象访问协议 (1.1):
<http://www.w3.org/TR/SOAP/>
- 简单对象访问协议 (1.2) 第 1 部分:
<http://www.w3.org/TR/soap12-part1/>
- 简单对象访问协议 (1.2) 第 2 部分:
<http://www.w3.org/TR/soap12-part2/>

了解 ASE Web 服务引擎

本章讨论以下内容：

主题	页码
Web 服务的生产者	13
Web 服务的使用者	16

ASE Web 服务引擎提供了以下功能：

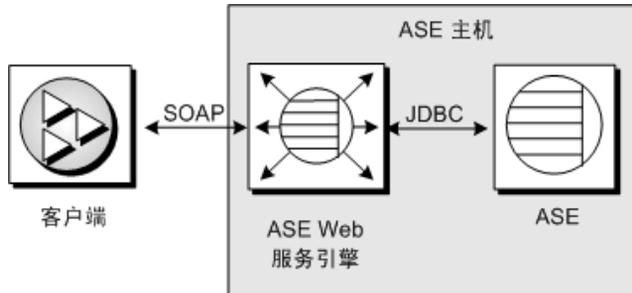
- [Web 服务的生产者](#) — 借助此功能，客户端应用程序可以使用 SOAP 访问 Adaptive Server Enterprise 存储过程和 SQL。
- [Web 服务的使用者](#) — 借助此功能，Adaptive Server Enterprise 可以访问和执行 Web 方法。

Web 服务的生产者

通过允许客户端应用程序使用 SOAP 访问 Adaptive Server Enterprise 存储过程和 SQL，ASE Web 服务引擎充当了 Web 服务的生产者。ASE Web 服务引擎的输出符合 SQLX。SQLX 被定义为 SQL 的 ANSI 规范的一部分。

注释 Sybase® 建议您让 ASE Web 服务引擎与 Adaptive Server Enterprise 在同一台计算机上运行。

图 2-1: 用于从客户端访问 ASE 的 ASE Web 服务引擎



客户端可以用 SOAP 请求的形式发送 SQL 或存储过程命令，所有结果都以 SOAP 响应的形式返回。SOAP 响应中的数据符合 SQLX 标准。

生产者组件

作为 Web 服务的生产者，ASE Web 服务引擎使用以下三个组件：HTTP 处理程序、SOAP 处理程序和 XML 映射程序。

图 2-2: 作为生产者的 ASE Web 服务引擎



HTTP 处理程序

HTTP 处理程序支持 HTTP 1.1，并监听使用 HTTP POST 和 GET 方法发送的请求。HTTP 处理程序还支持 SSL 连接。

注释 请勿使用 GET HTTP 请求。这些命令会将所有参数嵌入在 URL 中，使其无法进行加密。应使用 POST HTTP，它会将所有参数移动到 HTTP 请求的正文中，从而使全部内容都可以进行加密。

SOAP 处理程序

SOAP 处理程序支持 SOAP 1.2，并处理 SOAP 请求。SOAP 处理程序还生成描述 Web 服务的 WSDL 文件。

XML 映射程序

XML 映射程序将通过 JDBC 从 Adaptive Server Enterprise 返回的关系数据编码成符合 SQLX 标准的 XML。XML 映射程序还生成 DTD 和 XML 模式来描述数据。

生产者 Web 方法

ASE Web 服务引擎提供了以下方法：

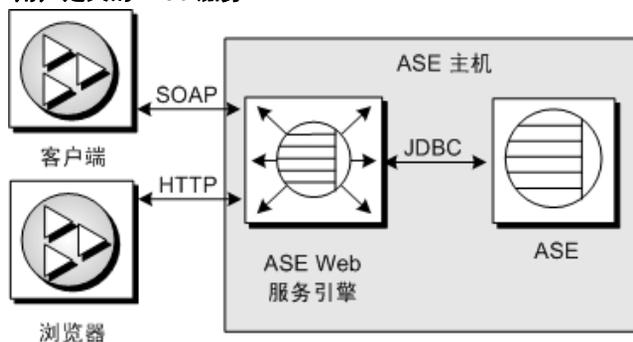
- `execute` — 执行 SQL 语句或存储过程。
- `login` — 建立与 Adaptive Server Enterprise 的持久连接。
- `logout` — 显式终止 Adaptive Server Enterprise 连接。

有关使用这些 Web 方法的信息，请参见第 4 章 “使用 ASE Web 服务”

用户定义的 Web 服务

除了由 ASE Web 服务引擎提供的 Web 方法外，ASE Web 服务还允许您使用 Web 浏览器或 SOAP 客户端在 Adaptive Server Enterprise 中创建 Web 服务并执行 SQL 命令。

图 2-3: 用户定义的 Web 服务



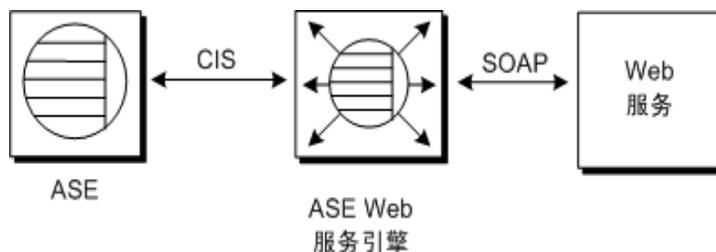
可以用 `create service` 命令创建用户定义的 Web 服务；使用 Web 服务，您可以指定要执行的 SQL、为可以用 `grant` 命令控制的权限创建一级对象以及控制该服务是否可以用 Web 浏览器或 SOAP 客户端调用。ASE Web Services Engine 为用户定义的 Web 服务自动生成 WSDL。有关创建和使用用户定义的 Web 服务的详细信息，请参见第 4 章“使用 ASE Web 服务”

注释 请勿使用 GET HTTP 请求。这些命令会将所有参数嵌入在 URL 中，使其无法进行加密。应使用 POST HTTP，它会将所有参数移动到 HTTP 请求的正文中，从而使全部内容都可以进行加密。

Web 服务的使用者

通过使 Adaptive Server Enterprise 可以访问和执行 Web 方法，ASE Web 服务引擎充当了 Web 服务的使用者。通过利用 Web 方法的 WSDL 文件中提供的信息，将 Web 方法映射到了 Adaptive Server Enterprise 代理表，使 Web 方法可供访问。然后，就可以对代理表使用 `select` 来调用 Web 方法。

图 2-4: 访问远程 Web 服务

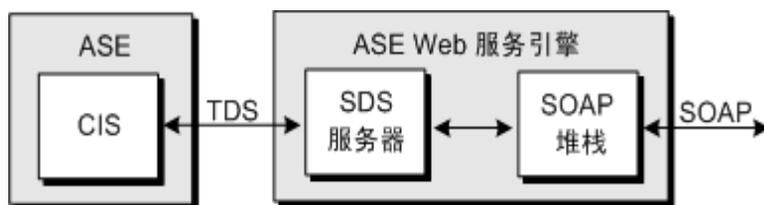


注释 所访问的 Web 服务可能处于防火墙之内或之外。

使用者组件

作为 Web 服务的使用者，ASE Web 服务引擎使用专业数据存储 (SDS) 服务器和 WSDL 分析程序。可以将 SDS 用作组件集成服务 (CIS) 以将 Web 方法映射到代理表。使用一个经 Apache Axis 分析过的 WSDL 文件构造代理表。

图 2-5: 用作使用者的 ASE Web 服务引擎



SDS 服务器

当 Adaptive Server Enterprise 收到针对一个 Web 方法代理表的 `select` 语句时，Adaptive Server Enterprise 将该请求通过 Tabular Data Stream™ (表式数据流，TDS) 转发给 ASE Web 服务引擎。SDS 服务器 (用作 CIS 的服务器) 将允许 ASE Web 服务引擎截取并处理来自 Adaptive Server Enterprise 的 TDS。

SOAP 堆栈

SOAP 堆栈是一组分层的功能，共同处理 XML 编码数据的序列化和传输。SOAP 堆栈利用 Web 方法的 WSDL 文件来确定相应代理表的结构。SOAP 堆栈还生成与提交到 Adaptive Server Enterprise 的 `select` 语句相对应的 SOAP 请求，并将这些 SOAP 请求发送到 SOAP 服务器。SOAP 堆栈同时支持 RPC/ 编码 Web 方法和文档 / 文字 Web 方法。

RPC/ 编码方法

在 RPC/ 编码 Web 方法的 SOAP 消息中，每个方法参数都有一个 XML 元素。RPC/ 编码 Web 方法的消息将根据 SOAP 规范进行编码。在表示 RPC/ 编码 Web 方法的代理表中，每个输入和输出参数都有一列。

注释 如果 RPC/ 编码 Web 方法没有输入或输出参数，则无法将其映射到代理表。没有参数的 Web 方法的代理表将不会有任何列，而在 Adaptive Server Enterprise 中是不能创建没有列的表的。

注释 当前，只将简单类型映射到列。在 RPC/ 编码 Web 方法中使用的复杂类型或数组导致无法将 Web 方法映射到代理表。

文档 / 文字方法

在文档 / 文字 Web 方法的 SOAP 消息中，通信方根据合并到 WSDL 文件中的 XML 模式指定所传输和格式化的数据。文档 / 文字 Web 方法的消息将根据 Web 方法的 WSDL 文件进行序列化和非序列化处理。表示文档 / 文字 Web 方法的代理表包含两列：`_inxml` 和 `outxml`。

制订实用互用性原则的 Web 服务互操作性 (WSI) 组织建议使用文档 / 文字 Web 方法来增强可移植性。Sybase 支持此建议。

代理表

由于 Web 服务代理表指向 Web 方法，因此它们是 `procedure` 类型的，并遵循该类型的所有限制：

- **命令**— 对于 `procedure` 类型的表，不能发出 `delete`、`update` 或 `insert` 命令，也不能发出 `create index`、`truncate table` 或 `alter table` 命令。
- **连接**— Web 服务代理表只能与另一个表连接，但该表不能是另一个 Web 服务代理表。
- **查询**— 以下划线（“_”）开头的列名用于指定输入参数。这些列被称为参数列，并且必须包含在 `select` 语句的 `where` 子句中。

有关对 `procedure` 表的限制的完整说明，请参见《组件集成服务用户指南》。

配置

本部分介绍如何配置 Adaptive Server 中的 Web 服务。



配置 ASE Web 服务

本章讨论以下内容：

主题	页码
配置	21
许可	23
配置文件	23
安全性	24

ASE Web 服务是作为 Adaptive Server Enterprise 安装的一部分安装的。可以在 Adaptive Server Enterprise 安装期间使用配置向导配置 ASE Web 服务，也可以在安装结束后通过 Sybase Central 进行配置。有关安装 Adaptive Server Enterprise 的说明，请参见 Adaptive Server Enterprise 《安装指南》。

注释 假定本章及随后各章所列的目录都位于 `$SYBASE/WS-15_0` 目录（对于 UNIX）和 `%SYBASE%\WS-15_0` 目录（对于 Windows）下，除非另作说明。

配置

可以在 ASE Web 服务安装期间或安装之后使用配置实用程序来配置 ASE Web 服务：

- [在安装期间进行配置](#) — 可以选择在 ASE Web 服务功能的自定义安装期间配置 ASE Web 服务。
- [在安装完成后进行配置](#) — 如果没有在安装期间配置 Web 服务功能，则必须在安装完成后从 Sybase Central 配置该功能。Sybase Central 提供了一个与安装过程中使用的配置实用程序相似的 GUI 配置实用程序。您也可以在无提示模式下或命令 / 主控台模式下配置 Web 服务功能。

在安装期间进行配置

要配置 Web 服务，请在 Web 服务功能安装期间激活配置实用程序，然后按照向导中的步骤操作。

在安装完成后进行配置

如果已经安装了 ASE Web 服务，则可以从 Sybase Central 中的 Adaptive Server Enterprise 插件启动配置实用程序。可以使用安装 GUI 配置 Web 服务，或者使用命令 / 主控台模式或无提示模式配置 Web 服务。

❖ 从 Sybase Central 启动配置实用程序 GUI

- 1 单击 Sybase Central 中 Adaptive Server Enterprise 插件的“文件夹”(Folders)视图中的“实用程序”(Utilities)文件夹。
- 2 找到“文件夹”(Folders)视图右边细节视图中的“配置 Web 服务”(Configure Web Service)图标。选择“配置 Web 服务”(Configure Web Service)图标以启动配置实用程序。
- 3 按照向导中的步骤操作。

❖ 在命令 / 控制台模式中启动配置实用程序

- 1 打开一个控制台窗口。
- 2 输入以下命令：

```
aseplugin -I
```
- 3 按照控制台上指示的步骤操作。

❖ 在无提示模式中启动配置实用程序

- 1 编辑 *myres.properties* 文件中的条目，指明需要使用的值。若要设置属性，请在 *myres.properties* 条目中添加“=”和属性值。有关 *myres.properties* 文件内容的详细信息，请参见[“myres.properties”](#)，它位于[附录“配置属性”](#)
- 2 打开一个控制台窗口。
- 3 输入以下命令：

```
aseplugin -s path
```

其中 *path* 是 *myres.properties* 文件的路径。

在无提示模式下，配置 Web 服务不需要其它的用户操作。

许可

ASE Web 服务的许可证输入是通过 Adaptive Server Enterprise 的 InstallShield 安装来处理的。

为从 SySAM 中检出 ASE Web 服务许可证，请在运行 ASE Web 服务引擎之前执行以下步骤：

- 1 与 Adaptive Server Enterprise 建立一个 isql 会话。
- 2 在 isql 中输入以下命令激活 ASE Web 服务：

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

配置文件

props 目录包括以下配置文件：

- [ws.properties](#) 文件
- [logging.properties](#) 文件
- [wsmsg.properties](#) 文件

ws.properties 文件

ws.properties 文件中包含 ASE Web 服务的配置设置。有关这些配置属性的详细信息，请参见附录 B “配置属性”

logging.properties 文件

logging.properties 文件定义向何处发送日志输出。可以将输出定向到日志文件或主控台。如果将输出定向到日志文件，则可以指定何时转滚到一个新日志文件。也可以控制记录消息的格式。有关日志转滚策略的信息，请参见第 50 页的“转滚日志文件”。

logging.properties 文件条目和日志行为遵照 Apache log4j 软件包的规则。有关详细信息，请参考 <http://jakarta.apache.org/log4j/docs/documentation.html> 上的 log4j 文档。

wsmmsg.properties 文件

wsmmsg.properties 文件供 ASE Web 服务在内部使用。

安全性

为确保 ASE Web 服务的安全运行， Sybase 建议执行以下步骤：

- 在安装 Adaptive Server Enterprise 的计算机上安装 ASE Web 服务。
- 使用 SSL 连接到 ASE Web 服务引擎。有关配置 SSL 的说明，请参见第 24 页的“配置 SSL”。

ASE Web 服务支持 Adaptive Server Enterprise 支持的所有授权测量。

注释 有关 ASE Web 服务安全性的最新信息，请参见 Adaptive Server Enterprise 《发行公告》。

配置 SSL

注释 安装期间将从 InstallShield 创建两个缺省的“sybase”证书口令。以后，可以从 Sybase Central 更改这两个口令。

可以从 InstallShield 或 Sybase Central 使用配置实用程序自动配置 SSL，但也可以手动配置 SSL。若要手动为 ASE Web 服务配置 SSL，请运行 *configssl* 脚本（此脚本位于 *bin* 目录中）：

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

其中：

- *domain_hostName* 是使用 SSL 连接到的 URL 的主机名。例如，以下 URL 的 *domain_hostName* 将会是 mydomainhostname：

```
http://mydomainhostname:8183/services/ase
```


此参数没有缺省值。
- *keystore* 是存储证书的位置和文件。对于 UNIX，缺省位置是 `$$SYBASE/WS-15_0/props/keystore`；对于 Windows，缺省位置是 `%SYBASE%\WS-15_0\props\keystore`。
- *httpsPort* 是用于监听 SSL 连接的端口。缺省值为 8182。
- *property_file* 是要更新的属性文件的位置和名称。对于 UNIX，缺省位置是 `$$SYBASE/WS-15_0/props/ws.properties`；对于 Windows，缺省位置是 `%SYBASE%\WS-15_0\props\ws.properties`。
- *certificate_password* 是证书的口令。此参数没有缺省值。如果调用脚本时没有提供口令，则脚本会提示输入一个值。
- *keystore_password* 是密钥存储口令。此参数没有缺省值。如果调用脚本时没有提供口令，则脚本会提示输入一个值。

注释 也可以添加自己的 SSL 证书。有关如何添加自己的证书的说明，请参见 JRE 中的 `keytool` 实用程序（用于处理 *keystore* 文件）的有关文档。随 ASE Web 服务一起提供的 JRE 的版本是 1.4。

WS.properties 文件使用 `com.sybase.ase.ws.producer.ssl.keystore` 设置 *keystore* 位置。WS 使用者是 *keystore* 位置的唯一用户。WS 使用者使用一个单独的证书，该证书在缺省情况下位于 JRE 中。若要以 WS 使用者的角色通过 HTTPS (SSL) 使用 Web 服务，请将生产者的 SSL 证书导入 JRE 的 *keystore*（例如，使用 `$$SYBASE/$SYBASE_JRE6_64/lib/security/cacerts`）。

为 Microsoft .NET 安装证书

Microsoft .NET 客户端需要相关证书才能使用 SSL 访问 ASE Web 服务引擎。可使用以下过程为 Microsoft .NET 安装证书。

❖ 为 Microsoft .NET 安装证书

- 1 用 SSL 启动 ASE Web 服务引擎。有关启动 ASE Web 服务引擎的说明，请参见第 29 页的“启动和停止 ASE Web 服务引擎”。
- 2 在 Microsoft Internet Explorer 的地址栏中输入以下地址：

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

其中：

- *producer_host* 是运行 ASE Web 服务引擎的主机。
- *SSL_port* 是供 ASE Web 服务引擎使用的端口。

出现将会“安全警报”对话框。

- 3 单击“查看证书”。出现“证书”对话框。
- 4 单击“安装证书”。证书管理器导入向导打开。
- 5 单击“下一步”，直到证书管理器导入向导指出证书已成功安装并返回“证书”对话框。
- 6 单击“确定”。浏览器返回到“安全警报”对话框。
- 7 单击“是”。浏览器窗口应显示标题为“Welcome to the ASE Web Services”（欢迎使用 ASE Web 服务）的网页。

❖ 检验证书的安装

- 1 关闭所有浏览器窗口。
- 2 重新启动 Microsoft Internet Explorer。
- 3 在 Microsoft Internet Explorer 的地址栏中输入以下地址：

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

其中：

- *producer_host* 是运行 ASE Web 服务引擎的主机。
- *SSL_port* 是供 ASE Web 服务引擎使用的端口。

此时，不应该出现“安全警报”对话框。

管理

本部分描述产品功能和管理任务。



本章讨论以下内容：

主题	页码
使用 ASE Web 服务引擎	29
使用用户定义的 Web 服务	40
ASE Web 服务日志记录	49
使用 Sybase Central	50

使用 ASE Web 服务之前，请确保已经完成第 3 章“配置 ASE Web 服务”中的配置任务。

使用 ASE Web 服务引擎

本节介绍以下内容：

- [启动和停止 ASE Web 服务引擎](#)
- [ASE Web 服务方法](#)
- [使用 sp_webservices](#)
- [调用 Web 服务](#)

启动和停止 ASE Web 服务引擎

要启动 ASE Web 服务的 ASE Web 服务引擎，请执行 `runws` 脚本，它位于 `bin` 目录中：

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

要停止 ASE Web 服务的 ASE Web 服务引擎，请执行 `stopws` 脚本，它也位于 `bin` 目录中：

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

runws 和 *stopws* 脚本的参数相同：

- *ase_username* 是 Adaptive Server Enterprise 的用户名。此参数没有缺省值。如果您没有给出此参数的值，系统将提示您输入一个值。
- *ase_password* 是 Adaptive Server Enterprise 的口令。此参数没有缺省值。如果您没有给出此参数的值，系统将提示您输入一个值。
- *ase_server_name* 是 Web 服务的名称。此参数没有缺省值。如果您没有给出此参数的值，系统将提示您输入一个值。
- *property_file* 是要更新的属性文件的位置和名称。对于 UNIX，缺省位置是 *\$\$SYBASE/WS-15_0/props/ws.properties*；对于 Windows，缺省位置是 *%SYBASE%\WS-15_0\props\ws.properties*。
- *-v* 指定在启动或关闭 ASE Web 服务引擎时显示版本信息。

情况

如果存在以下情况，将启动或停止 ASE Web 服务引擎：

- 所提供的 *ase_server_name* 位于 *libtcl.cfg* 文件指向的 LDAP 服务器上，或者位于 Adaptive Server Enterprise 的 *interfaces* 文件中。

ASE Web 服务首先在 *libtcl.cfg* 文件所指向的 LDAP 服务器上搜索包含 *ase_server_name* 的值的条目。ASE Web 服务使用 *ws.properties* 文件中的 *com.sybase.ase.ws.libtcl* 条目来查找 *libtcl.cfg* 文件。如果在 LDAP 服务器上未找到任何条目，ASE Web 服务将在 Adaptive Server Enterprise 的 *interfaces* 文件中查找条目。

注释 在 Windows 系统上，*interfaces* 文件的名称为 *sql.ini*。

ASE Web 服务使用 *ws.properties* 文件中的 *com.sybase.ase.ws.interfaces* 条目来查找 *interfaces* 文件。

- 使用所提供的 *ase_username* 和 *ase_password* 可以成功登录。

注释 Adaptive Server Enterprise 用户的口令不可以设置为空字符串。缺省情况下，*sa* 登录允许空字符串口令。Sybase 不建议使用空口令。

- 登录到 Adaptive Server Enterprise 的用户具有 *sa* 角色特权。
- 已在 Adaptive Server Enterprise 的 *isql* 中执行以下存储过程命令：

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

检验

成功执行 *runws* 脚本后，请检验 ASE Web 服务是否已启用以及 ASE Web 服务引擎是否正在运行。

❖ 检验 ASE Web 服务是否已启用

要检验 ASE Web 服务是否已启用，请执行以下操作：

- 在 Adaptive Server Enterprise 上执行以下命令：

```
sp_configure 'enable webservices'
```

如果 *sp_configure* 返回的值为 1，则表示 Web 服务功能已启用。如果返回的值为 0，则表示此功能未启用。

❖ 检验 ASE Web 服务引擎是否正在运行

- 检查 *logs* 目录中的 *producer.log* 或 *consumer.log* 文件，查找显示 ASE Web 服务引擎正在运行的消息。例如：

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

对于 SSL，日志中会显示 HTTPS 端口和相关的 SSL 信息。例如：

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

注释 为了兼容 ASE Web 服务的早期版本，15.0 版本中保留了 *runproducer*、*stopproducer*、*runconsumer* 和 *stopconsumer* 脚本。不过，在 15.0 版本中，这些脚本调用 *runws* 和 *stopws* 脚本。

ASE Web 服务方法

要访问 ASE Web 服务，客户端必须使用 ASE Web 服务引擎公开的方法。这些方法在 SOAP 中映射为 *rpc*：

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

已编码消息数据：

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

ASE Web 服务引擎提供了以下方法：

- **execute** — 执行 SQL 语句或存储过程。
- **login** — 建立与 Adaptive Server Enterprise 的持久连接。
- **logout** — 显式终止 Adaptive Server Enterprise 连接。

缺省情况下，Adaptive Server Enterprise 支持这些方法，并将这些方法作为 Web 服务来提供（通过一个 WSDL 文件）。无论使用 HTTP 调用还是使用 SSL 调用，这些方法的语法都一样。

execute

`execute` 方法用于在 Adaptive Server Enterprise 中执行 Transact-SQL 语句或存储过程。

语法

```
execute aseServerName userName password sqlxOptions sql
```

参数

- *aseServerName*

SOAP string，表示 *interfaces* 文件或 LDAP 服务器中 Adaptive Server Enterprise 服务器的名称。

在每次调用 `execute` 方法时，ASE Web 服务使用 *aseServerName* 值的方式与通过使用 *ase_server_name* 启动或停止 ASE Web 服务引擎的方式相同。有关详细信息，请参见第 29 页的“启动和停止 ASE Web 服务引擎”。

- *userName*

SOAP string，表示登录到 Adaptive Server Enterprise 所需的用户 ID。

- *password*

SOAP string，表示登录到 Adaptive Server Enterprise 所需的口令。

- *sqlxOptions*

SOAP string，表示一个或多个 `option` 参数。这些参数用于指定 SQLX 结果集的特性。以下是有效的选项参数：

- `general`
- `binary={hex | base64}`
- `columnstyle={element | attribute}`
- `entitize={yes | no | cond}`
- `format={yes | no}`
- `header={yes | no}`
- `multipleentitize={yes | no}`
- `multipleresults={all | data}`
- `ncr={non_ascii | no}`

- nullstyle={attribute | omit}
- prefix="value"
- root={yes | no}
- rowname="value"
- schemaloc="value"
- statement={yes | no}
- tablename="value"
- targetns="value"
- xsidecl={yes | no}

您必须为 *value* 提供值。有关 SQLX 函数和选项的详细信息，请参见《Adaptive Server Enterprise 中的 XML 服务》。

- *sql*

SOAP string 表示要在 Adaptive Server Enterprise 上执行的 SQL 语句或存储过程。在 *sql* 参数中指定的 SOAP 字符串大小要受到 *ws.properties* 文件中设置的 *com.sybase.ase.ws.maxpostsize* 属性值的限制。有关此属性和其它属性的说明，请参见附录 B “配置属性”

示例 1

此示例检查 Adaptive Server Enterprise 的版本号。

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

本示例直接调用 Web 方法。ASE Web 服务返回 XML 模式、DTD 以及包含已执行语句结果的结果集。

示例 2

本示例计算 pubs2 数据库中各个表的左连接。

```
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

login 方法可建立与 Adaptive Server Enterprise 的持久连接。

语法 `login aseServerName userName password`

参数

- *aseServerName*

SOAP string 表示要执行 SQL 语句或存储过程的 Adaptive Server Enterprise 的名称。

对于 login 方法, ASE Web 服务使用 *aseServerName* 值的方式与使用 *execute* 方法相同。

- *username*

SOAP string 表示登录到 Adaptive Server Enterprise 所需的用户 ID。

- *password*

SOAP string 表示登录到 Adaptive Server Enterprise 所需的口令。

用法

必须先建立一个连接, 然后才能在 Adaptive Server Enterprise 上执行 SQL 语句或存储过程。但是, login 方法是可选的。如果您在调用 *execute* 方法之前没有先调用 login 方法, 则 ASE Web 服务将自动建立与 Adaptive Server Enterprise 的非持久连接。login 方法可启动与 Adaptive Server Enterprise 的持久连接。使用 *logout* 方法可以终止连接。持久连接处于不活动状态的时间达到 60 秒后将自动终止。

logout

logout 方法用于终止与 Adaptive Server Enterprise 的持久连接。

语法 `logout`

用法

logout 方法用于终止由 login 方法建立的与 Adaptive Server Enterprise 的持久连接。

使用 sp_webservices

sp_webservices 存储过程创建并管理 ASE Web 服务引擎中使用的代理表。本节介绍 sp_webservices 的选项和参数。

sp_webservices 存储过程具有以下选项：

- **添加** — 创建代理表。
- **help** — 显示 sp_webservices 的用法信息。
- **list** — 列出映射到 WSDL 文件的代理表。
- **modify** — 修改超时设置。
- **remove** — 删除映射到 WSDL 文件的代理表。

此外，还有其它一些 sp_webservices 选项可用于用户定义的 Web 服务。有关这些选项的信息，请参见第 45 页的“将 sp_webservices 与用户定义的 Web 服务一起使用”。

注释 有关 Web 服务代理表限制的信息，请参见第 6 章“故障排除”中的“Web 服务代理表限制”。

添加

add 选项用于为 WSDL 文件指定的 Web 方法创建代理表。如果成功使用了 add 选项，将自动调用 list 选项以描述新代理表的模式。

语法

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

参数

- **wsdl_uri**
要映射到新代理表的 WSDL 文件的位置。如果指定了此参数，则 Web 服务会确保 syswsdl 表中存在 URI。
- **sds_name**
在 interfaces 或 sql.ini 文件中为 ASE Web 服务引擎指定的名称。缺省值为 ws。如果 sysattributes 表中不存在任何条目，则会产生错误。
- **method_name**
映射到代理表的 Web 方法的名称。所指定的 method_name 必须是在关联的 WSDL 文件中指定的 Web 方法的名称。
- **proxy_table**
在 method_name 中指定的 Web 方法映射到的代理表的名称。

用法

如果没有为 Web 方法指定 *method_name* 和 *proxy_table* 值，则缺省情况下为 Web 方法生成的代理表是在 WSDL 文件中指定的 Web 方法的名称。如果已存在一个具有此 Web 方法名称的代理表，则生成的新代理表的名称将类似如下形式：

```
method_nameN
```

其中 *method_name* 是缺省代理表名，*N* 是从 1 到 9 的数字，表示 Web 方法的连续映射。可以有最多 99 个重复代理表。

如果要为 Web 方法指定 *method_name* 和 *proxy_table* 值，则代理表的名称必须是新的。如果已存在一个具有 *proxy_table* 中指定了名称的代理表，则会产生错误，而且在 **add** 选项中指定的所有 Web 方法都不会映射到代理表。

add 选项的输出中列出了已成功映射到代理表的方法，以及那些尚未映射的方法。未映射的 Web 方法的代理表名称在 **add** 选项的输出中以 NULL 表示。

注释 对于为 RPC/ 编码 Web 方法和文档 / 文字 Web 方法生成的代理表来说，用于输入和输出的列是不同的。在表示 RPC/ 编码 Web 方法的代理表中，每个输入和输出参数都有对应的一列。表示文档 / 文字 Web 方法的代理表包含 *_inxml* 和 *outxml* 两列。

注释 有关数据类型映射的信息，请参见附录 C “SOAP 和 Adaptive Server Enterprise 的数据类型映射”

help

help 选项提供了阐明如何使用 `sp_webservices` 存储过程的说明和示例。

语法

```
sp_webservices help [, 'option']
```

参数

- *option*

要向其提供详细选项的选项。有效值包括 **add**、**list**、**remove** 和 **modify**。

用法

如果没有为 *option* 指定任何值，则 **help** 选项将输出 **add**、**addalias**、**deploy**、**dropalias**、**list**、**listalias**、**listudws**、**modify**、**remove** 和 **undeploy** 选项的简短语法描述。

list

`list` 选项用于列出 WSDL 文件中描述的 Web 方法。

语法

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

参数

- *wsdl_uri*
所映射的 WSDL 文件的 URI。如果没有为 *wsdl_uri* 指定值，则 `list` 选项将显示有关已经映射到代理表的所有 Web 方法的信息。
- *sds_name*
在 *interfaces* 或 *sql.ini* 文件中为 ASE Web 服务引擎指定的 SDS 服务器的名称。缺省值为 `ws`。如果 `sysattributes` 表中不存在任何条目，则会产生错误。

如果 *wsdl_uri* 和 *sds_name* 这两个参数都没有指定，则会按 `wsdlid` 顺序列出 `sysattributes` 表中的所有条目。

用法

如果 WSDL 文件中描述的 Web 方法已经映射到代理表，则 `list` 选项将输出有关各个代理表的信息。如果 WSDL 文件中描述的 Web 方法没有映射到代理表，则 `list` 选项将输出可用于创建代理表的 SQL。

modify

`modify` 选项用于修改 WSDL 文件的属性信息。

语法

```
sp_webservices 'modify', 'wsdl_uri', 'timeout=time'
```

参数

- *wsdl_uri*
要为其更改属性信息的 WSDL 文件的 URI。
- *time*
操作中止之前，Web 方法必须响应的时间间隔（以秒为单位）。

remove

`remove` 选项用于删除 Web 方法的代理表映射。

语法

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

参数

- *wsdl_uri*
要删除其代理表的 WSDL 文件的 URI。
- *sds_name*
在 *interfaces* 或 *sql.ini* 文件中为 ASE Web 服务引擎指定的 SDS 服务器的名称。缺省值为 `ws`。如果 `sysattributes` 表中不存在任何条目，则会产生错误。

调用 Web 服务

若要使用 ASE Web 服务引擎调用 Web 服务，请按以下步骤操作：

❖ 调用 Web 服务

- 1 启动 ASE Web 服务引擎。
- 2 使用 `sp_webservices` 的 `add` 选项将 Web 服务映射到 Adaptive Server Enterprise 的代理表中。
- 3 使用 `sp_help` 确定调用 Web 方法所需的输入和输出参数。
- 4 通过对代理表使用 `select` 语句来调用 Web 方法。

示例 1

本示例调用 RPC/ 编码 Web 方法，以显示两个并发之间的交换率。

使用 `sp_webservices` 的 `add` 选项将 Web 方法映射到代理表：

```
1> sp_webservices 'add',  
'http://www.xmethods.net/sd/2001/CurrencyExchangeService.wsdl'  
2> go
```

Web 方法 `getRate` 将映射到具有相同名称的代理表。

从代理表选择该 Web 方法以调用该 Web 方法：

```
1> select * from getRate where _country1 = 'usa' and _country2 = 'india'  
2> go
```

以前的 `select` 返回的结果显示指定的参数的交换率：

```
Result          _country1      _country2  
43.000000      usa            india  
(1 row affected)
```

示例 2

本示例调用 Web 方法以显示 XML 文档中的库存信息。

使用 `sp_webservices` 的 `add` 选项将 Web 方法映射到代理表：

```
1> sp_webservices "add" , "http://www.webservicex.net/stockquote.asmx?WSDL"  
2> go
```

Web 方法 `GetQuote1` 将映射到与其同名的代理表。

通过选择 GetQuote1 代理表的 outxml 列可调用 Web 方法:

```
1> select outxml from GetQuote1 where _inxml = '<?xml version="1.0"
encoding="utf-8"?>
2>     <GetQuote1 xmlns="http://www.webserviceX.NET/">
3>         <symbol>SY</symbol>
4>     </GetQuote1>'
5> go
```

前面的 select 返回的结果显示 XML 文档中的报价信息:

```
outxml

<?xml version="1.0" encoding="UTF-8" ?><GetQuote1Response
xmlns="http://www.webserviceX.NET/"><GetQuoteResult><StockQuotes><Stock><Symbo
ol>SY</Symbol><Last>21.48</Last><Date>7/21/2005</Date><Time>4:01pm</Time><Cha
nge>+1.72</Change><Open>20.00</Open><High>21.60</High><Low>19.91</Low><Volume
>2420100</Volume><MktCap>1.927B</MktCap><PreviousClose>19.76</PreviousClose><
PercentageChange>+8.70%</PercentageChange><AnnRange>12.75 -
20.44</AnnRange><Earnings>0.706</Earnings><P-E>27.99</P-E><Name>SYBASE
INC</Name></Stock></StockQuotes></GetQuoteResult></GetQuote1Response>

(1 row affected)
```

示例 3

本示例通过一个视图来调用映射到上一示例中的代理表的 GetQuote1 Web 方法, 从而显示股票信息。

要使用此 Web 服务, 必须创建一个表来保存表示股票的符号:

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

向 stocksymbol 表中插入数据:

```
1> insert stocksymbol values("SY")
2> insert stocksymbol values("ORCL")
3> go
```

现在创建一个调用 `GetQuote1` Web 方法的视图:

```
1> CREATE VIEW getstockvw as
2> select Symbol = xmlextract('//Stock/Symbol/text()',outxml returns
varchar(5)),
3>   Name = xmlextract('//Stock/Name/text()',outxml returns varchar(20)),
4>   Time = xmlextract('//Stock/Time/text()',outxml returns varchar(10)),
5>   Date = xmlextract('//Stock/Date/text()',outxml returns date),
6>   High = xmlextract('//Stock/High/text()',outxml returns decimal(15,2)),
7>   Low = xmlextract('//Stock/Low/text()',outxml returns decimal(15,2))
8> FROM GetQuote1 ,stocksymbol
9> WHERE _inxml = '<GetQuote1
xmlns="http://www.webserviceX.NET/"><symbol>'+symbol+'</symbol></GetQuote1>'
10> go
```

从 `getstockvw` 视图中进行选择以查看 `GetQuote1` 方法的输出:

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

前面的 `select` 返回的结果显示视图定义指定的参数的报价信息:

Symbol	Name	Time	Date	High	Low
SY	SYBASE INC	4:01pm	Jul 21 2005	21.60	19.91
ORCL	ORACLE CORP	4:00pm	Jul 21 2005	14.05	13.54
MSFT	MICROSOFT CP	4:00pm	Jul 21 2005	26.48	26.19

(3 rows affected)

使用用户定义的 Web 服务

本节描述了特定于用户定义的 Web 服务的功能, 包含以下主题:

- 用户定义的 Web 服务的命令
- 将 `sp_webservices` 与用户定义的 Web 服务一起使用
- 用户定义的 Web 服务的安全性
- 用户定义的 Web 服务的审计

用户定义的 Web 服务的命令

可以使用以下命令创建、删除和变更用户定义的 Web 服务：

- [create service](#)
- [drop service](#)

create service

`create service` 命令使用指定的名称和参数将所提供的 SQL 语句包装到存储过程中。除以下不同外，结果存储过程的行为与使用 `create procedure` 命令创建的存储过程相同（遵循执行、复制、`sp_helptext` 和重新编译等所依据的现有存储过程规则，且可使用 `isql` 来执行）：

- 结果存储过程仅可用 `drop service` 命令删除，而不能用 `drop procedure` 命令删除。
- 将使用重新创建 `create service` 命令所需的 DDL 来填充 `syscomments` 表。
- 指定的服务名称不能创建存储过程组。

注释 要想能够通过 ASE Web 服务引擎获得用户定义的 Web 服务，必须使用 `sp_webservices` 的 `deploy` 选项。然而，即使没有配置，也可以从 `isql` 访问用户定义的 Web 服务的存储过程。有关 `sp_webservices` 的 `deploy` 选项的信息，请参见第 45 页的“[deploy](#)”（在“[使用 sp_webservices](#)”中）。

语法

```
create service service-name [secure security_options ] [, userpath path]
    [, alias alias-name]
    type { xml | raw | soap }
    [[(@parameter_name datatype [(length ) | (precision [, scale ])]
      [= default][output]
    [, @parameter_name datatype [(length ) | (precision [, scale ])]
      [= default][output]...[...])]
as SQL_statements
security_options ::= ( security_option_item [security_option_item])
```

参数

- *service-name* — 用户定义的 Web 服务的名称。该名称可以是任何对存储过程有效的名称。当使用该服务名称调用 `drop service` 命令时，将删除相应的存储过程。如果指定现有服务的名称，则会引发异常。
- *security_option_item* — 为 `clear` 或 `ssl`：
 - `clear` 表示使用 HTTP 访问此 Web 服务。
 - `ssl` 表示使用 HTTPS 访问此 Web 服务。

- *path* — 字符串文字，指定要附加到用于访问 Web 服务的 URL 的用户定义路径。缺省状态下，此路径为空值。
- *alias-name* — 字符串文字，用于指定用户定义的 Web 服务别名。
- *parameter_name* — 用户定义的 Web 服务的参数名称。Web 服务执行时，需提供该参数的值。参数名前面必须有 @ 符号且必须符合标识符的规则。这些情况与 `create procedure` 命令的 *parameter_name* 参数相同。
- *SQL_statements* — 用户定义的 Web 服务要采取的操作。可以包含任何数量和任何种类的 SQL 语句，但 `create view`、`create default`、`create rule`、`create procedure`、`create trigger` 和 `use` 除外。这些情况与 `create procedure` 命令的 *SQL_statements* 参数相同。
- *type* — 可以是 `soap`、`raw` 或 `xml`:
 - `soap` 类型意味着使用 HTTP POST 请求，同时还意味着必须符合所有的 SOAP 规则。数据以 SQL/XML 格式返回。
 - `raw` 类型表示在不需要变更或重新格式化的情况下发送输出。这意味着使用 HTTP GET 请求。调用的存储过程可以指定精确的输出。
 - `xml` 类型表示结果集输出以 SQL/XML 格式返回。这意味着使用 HTTP GET 请求。

注释 有关 ASE 存储过程和 SOAP 用户定义的 Web 服务之间的数据类型映射，请参见附录 C “SOAP 和 Adaptive Server Enterprise 的数据类型映射” 中的 “`create service` 命令的 ASE 到 SOAP 数据类型映射”。

示例 1

此示例中创建了一个类型为 `raw` 的、用户定义的 Web 服务 `rawservice` 以返回当前数据库版本。从 `pubs2` 数据库的 `isql` 命令行输入了 `create service` 命令：

```
1> use pubs2
2> go
1> create service rawservice type raw as select
'<html><h1>' + @@version + '</h1></html>'
2> go
```

然后必须配置新创建的用户定义的 Web 服务：

```
1> sp_webservices 'deploy', 'all'
2> go
```

注释 有关 `sp_webservices` 的 `deploy` 选项的详细信息，请参见第 45 页的“将 `sp_webservices` 与用户定义的 Web 服务一起使用”。

新创建的用户定义 Web 服务的 WSDL 可以在以下 URL 处找到：

```
http://myhost:8181/services/pubs2?wsdl
```

新创建的用户定义 Web 服务可在以下 URL 处找到：

```
http://myhost:8181/services/pubs2?method=rawservice&
username=bob&password=bob123
```

其中，`bob` 和 `bob123` 是用户定义 Web 服务的创建者的用户 ID 和口令。

输出（Adaptive Server Enterprise 版本字符串）将显示在浏览器窗口中的 HTML `<h1>` 标记中。

示例 2

此示例中创建了一个类型为 `xml` 的、用户定义的 Web 服务 `xmlservice` 以返回当前数据库版本。从 `pubs2` 数据库的 `isql` 命令行输入了 `create service` 命令：

```
1> use pubs2
2> go
1> create service xmlservice userpath "testing" type xml
as select @@version
2> go
```

然后必须配置新创建的用户定义的 Web 服务：

```
1> sp_webservices 'deploy', 'xmlservice'
2> go
```

注释 有关 `sp_webservices` 的 `deploy` 选项的详细信息，请参见第 45 页的“将 `sp_webservices` 与用户定义的 Web 服务一起使用”。

用户定义的 Web 服务的 WSDL 可以在以下 URL 处找到：

```
http://myhost:8181/services/pubs2/testing?wsdl
```

使用浏览器，可以在以下 URL 处调用用户定义的 Web 服务：

```
http://myhost:8181/services/pubs2/testing?method=xmlservice&
username=bob&password=bob123
```

其中，`bob` 和 `bob123` 是用户定义 Web 服务的创建者的用户 ID 和口令。输出以 XML 格式显示在浏览器窗口中。

示例 3

此示例中向 SOAP 客户端提供了用户定义的 Web 服务以执行存储过程 `sp_who`。提供了一个参数，且指定了可选的 `userpath` 标识：

```
create service sp_who_service userpath
'myservices/args' type soap @loginname varchar(30) as
exec sp_who @loginname
```

这里创建了 Web 服务作为 `pubs2` 数据库中的 `sp_who_service`。在配置之后，可以从以下 URL 处访问它：

`http://localhost:8181/pubs2/myservices/args/sp_who_service`

该服务的 WSDL 可以在以下 URL 处获得：

`http://localhost:8181/pubs2/myservices/args?wsdl`

WSDL 文件中描述的 Web 方法的签名如下：

```
DataReturn[] sp_who_service (xsd:string username,
xsd:string password, xsd:string loginname)
```

新服务是由具有参数 `loginname`（其类型为 `varchar(30)`）的 SOAP 客户端调用的。

drop service

`drop service` 命令用于从当前数据库中删除用户定义的 Web 服务。元数据和对应的存储过程都将被删除。

注释 您必须先取消配置用户定义的 Web 服务，然后才能将其删除。有关 `sp_webservices` 的 `undeploy` 选项的详细信息，请参见第 45 页的“[将 sp_webservices 与用户定义的 Web 服务一起使用](#)”。

语法

```
drop service service-name
```

参数

- *service-name* — 用户定义的 Web 服务的名称。该名称可以是任何对存储过程有效的名称。如果指定了一个不存在的服务名称，则会产生异常。此外，您也不能删除其它会话当前正在使用的服务。

示例

此示例删除名为 `sp_who_service` 的用户定义的 Web 服务：

```
drop service sp_who_service
```

将 sp_webservices 与用户定义的 Web 服务一起使用

以下 sp_webservices 选项用于用户定义的 Web 服务：

- **addalias** — 创建数据库别名。
- **deploy** — 配置用户定义的 Web 服务。
- **dropalias** — 删除数据库别名。
- **listudws** — 列出映射到 WSDL 文件的代理表。
- **listalias** — 列出一个或多个数据库别名。
- **undeploy** — 取消对用户定义的 Web 服务的配置。

addalias

add alias 选项用于创建表示数据库名称的别名。

语法

```
sp_webservices 'addalias' alias_name , database_name
```

参数

- *alias_name*

指定的数据库的别名。该参数是必需的。

- *database_name*

为其指定了别名的数据库的名称。该参数是必需的。

用法

使用别名可以在指定 URL 中表示数据库名称的部分时进行更好的控制。将别名与 create service 命令的 userpath 部分搭配使用时，可以完全控制用于访问用户定义的 Web 服务的 URL。

deploy

deploy 选项用于配置用户定义的 Web 服务，使 ASE Web 服务引擎可通过 HTTP 或 HTTPS 对其进行访问。

注释 请勿使用 GET HTTP 请求。这些命令会将所有参数嵌入在 URL 中，使其无法进行加密。应使用 POST HTTP，它会将所有参数移动到 HTTP 请求的正文中，从而使全部内容都可以进行加密。

语法

```
sp_webservices 'deploy', ['all' | 'service_name']
```

参数	<ul style="list-style-type: none">• <code>all</code> 指定为当前数据库配置所有用户定义的 Web 服务。• <code>service_name</code> 要配置的用户定义的 Web 服务的名称。
用法	<p><code>deploy</code> 和 <code>undeploy</code> 选项用于控制用户定义的 Web 服务何时可用。此选项需要 <code>webservices_role</code> 特权。</p> <p>如果指定了 <code>all</code> 参数，则 ASE Web 服务引擎将删除其用户定义的 Web 服务的内部高速缓存，并从 Adaptive Server Enterprise 中重新读取有关用户定义的 Web 服务的所有元数据。</p> <hr/> <p>注释 不能删除或重命名当前配置的用户定义的 Web 服务。</p> <hr/>

dropalias

`dropalias` 选项用于删除表示数据库名称的别名。

语法 `sp_webservices 'dropalias' alias_name`

参数

- `alias_name`
要删除的别名。

用法 如果一个已配置的用户定义的 Web 服务正在引用一个别名，则无法删除该别名。要删除该别名，必须首先取消配置引用该别名的用户定义的 Web 服务。

listudws

`listudws` 选项用于列出当前数据库的用户定义的 Web 服务。

语法 `sp_webservices 'listudws' [, 'service_name']`

参数

- `service_name`
要列出的用户定义的 Web 服务的名称。

用法 如果未指定 `service_name` 参数，则将列出所有用户定义的 Web 服务。

listalias

`listalias` 选项用于列出所有别名。

语法 `sp_webservices 'listalias'`

用法 列出所有别名。

undeploy

`undeploy` 选项用于使 ASE Web 服务引擎无法通过 HTTP 或 HTTPS 访问用户定义的 Web 服务。

语法 `sp_webservices 'undeploy', ['all' | 'service_name']`

参数

- `all`
指定为当前数据库取消配置所有用户定义的 Web 服务。
- `service_name`
要取消配置的用户定义的 Web 服务的名称。

用法 `deploy` 和 `undeploy` 选项用于控制用户定义的 Web 服务何时可用。此选项需要 `webservices_role` 特权。

用户定义的 Web 服务的安全性

Adaptive Server Enterprise 中添加了系统角色 `webservices_role`，用于实现 Web 服务功能。在使用 `sp_webservices` 的 `deploy` 和 `undeploy` 选项时，此角色是必不可少的。要执行用户定义的 Web 服务，必须具有用于执行相应存储过程的有效登录名和权限。

要创建、删除和执行用户定义的 Web 服务，需要具有在 Adaptive Server Enterprise 中创建、删除和执行存储过程所需的特权。有关如何使用 `grant` 和 `revoke` 命令设置适当特权的详细信息，请参见 Adaptive Server Enterprise 《安全性管理指南》。

注释 有关 ASE Web 服务安全性的最新信息，请参见 Adaptive Server Enterprise 《发行公告》。

用户定义的 Web 服务的审计

可以将用户定义的 Web 服务作为 Adaptive Server Enterprise 中的存储过程建模。在处理用户定义的 Web 服务过程中，Adaptive Server Enterprise 将使用现有审计范围为存储过程

生成以下事件：

- 创建用户定义的 Web 服务 — 生成名为 “Create Procedure” 的事件 11
- 删除用户定义的 Web 服务 — 生成名为 “Drop Procedure” 的事件 28
- 执行用户定义的 Web 服务 — 生成名为 “Execution of Stored Procedure” 的事件 38

有关现有审计功能的详细信息，请参见 Adaptive Server Enterprise 《安全管理指南》。

除现有审计功能外，Adaptive Server Enterprise 还为 `sp_webservices` 的 `deploy` 和 `undeploy` 选项提供了两个审计事件。

审计记录存储在 `sysaudits` 系统表中。可以使用以下命令对 Web 服务启用审计：

```
sp_audit "security", "all", "all", "on"
```

审计 `sp_webservices 'deploy'`

审计事件号 110 对应于 `sp_webservices` 的 `deploy` 选项。

示例 1

此示例说明由用户 `bob` 在 `pubs2` 数据库中输入的以下命令的审计表条目：

```
sp_webservices 'deploy', 'all'
```

对应的审计表条目列出了 110、`bob` 和 `pubs2`，分别作为 `event`、`loginname` 和 `dbname` 列中的值。`extrainfo` 列包含：

```
webservices_role; deploy_all; ; ; ; bob/ase;
```

示例 2

此示例说明由用户 `bob` 在 `pubs2` 数据库中输入的以下命令的审计表条目：

```
sp_webservices 'deploy', 'rawservice'
```

对应的审计表条目列出了 110、`bob` 和 `pubs2`，分别作为 `event`、`loginname` 和 `dbname` 列中的值。`extrainfo` 列包含：

```
webservices_role; deploy; ; ; ; bob/ase;
```

有关 `sysaudits` 表列的完整说明，请参见 Adaptive Server Enterprise 《安全管理指南》。

审计 sp_webservices 'undeploy'

审计事件号 111 对应于 sp_webservices 的 undeploy 选项。

示例 1 此示例说明由用户 bob 在 pubs2 数据库中输入的以下命令的审计表条目:

```
sp_webservices 'undeploy', 'all'
```

对应的审计表条目列出了 111、bob 和 pubs2，分别作为 event、loginname 和 dbname 列中的值。extrainfo 列包含:

```
webservices_role; undeploy_all; ; ; ; bob/ase;
```

示例 2 此示例说明由用户 bob 在 pubs2 数据库中输入的以下命令的审计表条目:

```
sp_webservices 'undeploy', 'rawservice'
```

对应的审计表条目列出了 111、bob 和 pubs2，分别作为 event、loginname 和 dbname 列中的值。extrainfo 列包含:

```
webservices_role; deploy; ; ; ; bob/ase;
```

有关 sysaudits 表列的完整说明，请参见 Adaptive Server Enterprise 《安全性管理指南》。

ASE Web 服务日志记录

缺省情况下，ASE Web 服务仅记录信息性消息和错误消息。有关如何记录更多详细信息的详情，请联系 Sybase 技术支持部门。

本节介绍以下主题:

- [ASE Web 服务日志文件](#)
- [转滚日志文件](#)

ASE Web 服务日志文件

ASE Web 服务将活动记录到以下三个文件中:

- [consumer.log](#)
- [producer.log](#)
- [http.log](#)

consumer.log

consumer.log 文件包含由通过代理表调用远程 Web 服务的 Adaptive Server Enterprise 产生的信息和错误消息。

producer.log

producer.log 文件包含由 Adaptive Server Enterprise 接收的 SOAP 请求所产生的所有信息和错误消息。

http.log

http.log 文件以 NCSA 请求日志格式保存所有 HTTP 请求。调用的每个 Web 方法都存在一个 HTTP 请求。

转滚日志文件

在 ASE Web 服务中，日志记录是使用 Apache log4j 框架实现的。有关特定的 log4j 参数以及如何实现转滚策略的信息，请参见以下 Web 文档：
<http://jakarta.apache.org/log4j/docs/>。

使用 Sybase Central

使用 ASE Web 服务的 Sybase Central 插件可以执行以下任务：

- 从 Sybase Central 中配置 Web 服务
- 使用 Web 服务
- 执行 Web 服务
- 删除 Web 方法
- 添加用户定义的 Web 服务
- 添加别名
- 显示 Web 方法的属性

有关详细信息，请参见 Sybase Central 中有关 ASE Web 服务的联机帮助。

示例应用程序

本章讨论以下内容：

主题	页码
Apache 示例客户端	51
Microsoft .NET 示例客户端	55

在 *samples* 目录下提供了可用于创建和运行示例客户端的工具。本章介绍为 ASE Web 服务引擎提供的示例应用程序。

Apache 示例客户端

本节介绍位于 `$SYBASE/WS-15_0/samples/apacheclient` 目录中（UNIX 上）或位于 `%SYBASE%\WS-15_0\samples\apacheclient` 目录中（Windows 上）的示例客户端和脚本。

注释 如果打算在没有安装 ASE Web 服务的计算机上运行 Apache 示例客户端，必须将 `/apacheclient/lib` 目录的内容复制到该计算机上。

创建示例客户端

要使用所提供的示例脚本，必须首先创建示例客户端。

❖ 创建示例客户端

- 如有必要，请更改 `apacheclient` 目录中所有脚本中的变量定义，确保 `JRE` 变量指向您的 `JRE`。必须使用 `JRE 1.4.2` 版或更高版本。缺省情况下，使用 `$SYBASE_JRE` 目录（UNIX 上）或 `%SYBASE_JRE%` 目录（Windows 上）中提供的 `JRE`。

创建 ASE Web 服务客户端后，便可以运行示例脚本来执行存储过程和 SQL 语句。该脚本位于 `apacheclient` 目录中。

使用 runexecute

runexecute 脚本通过 ASE Web 服务在 Adaptive Server Enterprise 上执行存储过程或 Transact-SQL 语句。该示例应用程序调用 `execute` Web 方法。

语法

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

参数

- *web_service_URL*
正在使用的 Web 服务的位置。
- *aseServerName*
SOAP string, 表示 *interfaces* 文件或 LDAP 服务器中 Adaptive Server Enterprise 服务器的名称。
- *user_ID*
登录到 Adaptive Server Enterprise 所需的用户 ID。
- *password*
登录到 Adaptive Server Enterprise 所需的口令。
- *SQLX_option*
表示一个或多个 `option` 参数的字符串。这些参数用于指定 SQLX 结果集的特性。以下是有效的选项参数：
 - `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"`

必须为 *value* 提供值。有关 SQLX 函数和选项的详细信息，请参见《Adaptive Server Enterprise 中的 XML 服务》。

- *output_class*

所需的输出类型。以下是此参数的有效值：

- *schema* — 返回 XML 模式。
- *dtd* — 返回 XML DTD。
- *data* — 返回结果集。
- *all* — 返回模式、DTD 和数据。

- *count*

执行此语句的次数。如果 *count* 的值大于 1，则会创建一个会话并使用持久连接。

- *sql_statement*

要在 Adaptive Server Enterprise 上执行的语句。该语句必须用双引号分隔。

示例 1

此示例使用 **select** 语句检查 Adaptive Server Enterprise 的版本号。

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

ASE Web 服务返回 XML 模式、DTD 以及包含已执行语句结果的结果集。

示例 2

下面的示例对 **pubs2** 数据库执行名为 **booksales** 的存储过程。该存储过程将返回已售出的具有指定书名 ID 的书籍数量。

```
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"
```

ASE Web 服务返回 XML 模式、DTD 以及包含已执行语句结果的结果集。

返回的结果集如下：

```
<?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>
```

返回的 DTD 如下:

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

返回的模式如下:

```
<?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="Unnamedcoll"  
      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:maxLength 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:maxLength value="24"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name="resultset"
  type="TableType.resultset"/>
</xsd:schema>

```

示例 3

下面的示例对 **pubs2** 数据库执行一个 SQL 查询。该查询将返回不与出版者住在同一城市的作者的姓氏和所住城市。

```

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)"

```

ASE Web 服务返回包含已执行语句结果的结果集。

Microsoft .NET 示例客户端

本节介绍位于 `$SYBASE/WS-15_0/samples/ms.net/Execute/bin/Release` 目录中（UNIX 上）或位于 `%SYBASE%\WS-15_0\samples\ms.net\Execute\bin\Release` 目录中（Windows 上）的示例客户端和脚本。

可以通过下面的 URL 下载 Microsoft .NET:

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

注释 该 URL 至少到本文档出版之日是可用的，但可能会随时间而改变。

创建示例客户端

要使用所提供的示例脚本，必须首先创建示例客户端。创建示例客户端之后，就可以运行本节中介绍的示例脚本。该脚本位于 *Release* 目录中。

使用 Execute.exe

Execute.exe 可以通过 ASE Web 服务在 Adaptive Server Enterprise 上执行存储过程或 Transact-SQL 语句。该示例应用程序调用 *execute* Web 方法。

语法

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

参数

- *web_service_URL*
正在使用的 Web 服务的位置。
- *aseServerName*
SOAP string，表示 *sql.ini* 文件或 LDAP 服务器中 Adaptive Server Enterprise 服务器的名称。
- *user_ID*
登录到 Adaptive Server Enterprise 所需的用户 ID。
- *password*
登录到 Adaptive Server Enterprise 所需的口令。
- *SQLX_option*
表示一个或多个 *option* 参数的字符串。这些参数用于指定 SQLX 结果集的特性。以下是有效的选项参数：
 - *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"`

必须为 *value* 提供值。有关 SQLX 函数和选项的详细信息，请参见《Adaptive Server Enterprise 中的 XML 服务》。

- *output_class*

所需的输出类型。以下是此参数的有效值：

- `schema` — 返回 XML 模式。
- `dtd` — 返回 XML DTD。
- `data` — 返回结果集。
- `all` — 返回模式、DTD 和数据。

- *count*

执行的次数。

- *sql_statement*

要在 Adaptive Server Enterprise 上执行的语句。该语句必须用双引号分隔。

示例 1

此示例检查 Adaptive Server Enterprise 的版本号。

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

ASE Web 服务返回 XML 模式、DTD 以及包含已执行语句结果的结果集。

示例 2

下面的示例对 `pubs2` 数据库执行名为 `booksales` 的存储过程。该存储过程将返回已售出的具有指定书名 ID 的书籍数量。

```
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"
```

ASE Web 服务返回 XML 模式、DTD 以及包含已执行语句结果的结果集。

示例 3

下面的示例对 **pubs2** 数据库执行一个 SQL 查询。该查询将返回不与出版者住在同一城市的作者的姓氏和所住城市。

```
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)"
```

ASE Web 服务返回包含已执行语句结果的结果集。

故障排除

本部分描述故障排除过程和错误消息。

本章讨论以下内容：

主题	页码
故障排除问题	61
诊断工具	66
消息	68

故障排除问题

以下问题可以帮助您排除 ASE Web 服务的故障。

远程服务器类定义设置

问题

`sp_webservices add` 命令在生成代理表时可能返回以下错误：

```
Warning: Row size (3347 bytes) could exceed row size limit, which is 1962
bytes.
Msg 208, Level 16, State 1:
Server 'JMALVARADO', Line 1:
tempdb..ws_4338e6e122cd4ef0a not found. Specify owner.objectname or uses to
check whether the object exists (sp_help may produce lots of output).
No proxy tables were created for the WSDL URL:
[http://www.xignite.com/xquotes.asmx?WSDL]
(return status = 0)
```

出现此错误是因为使用 `sp_addserver` 并使用一个非“sds”的类添加了表示 ASE Web 服务引擎的远程服务器。要检验是不是这种情况，请使用 `isql` 中的 `sp_helpserver`：

```
sp_helpserver ws
```

此处的 *ws* 是 ASE Web 服务引擎的名称。这是缺省值。远程服务器类将在如下所示的结果列中返回：

```
name network_name class ...
---- -
ws ws null ...
```

用户操作

通过使用 `isql` 中的 `sp_dropserver` 和 `sp_addserver` 将远程服务器的类更改为 “sds”：

```
sp_dropserver ws_name
...
sp_addserver ws_name, sds, ws_name
```

此处的 *ws_name* 是为 ASE Web 服务引擎选择的名称。

未映射 RPC/ 编码 Web 方法

问题

如果 RPC/ 编码 Web 方法没有输入或输出参数，则无法将其映射到代理表。没有参数的 Web 方法的代理表将不会有任何列，而在 Adaptive Server Enterprise 中是不能创建没有列的表的。

用户操作

修改 Web 方法，在其中包含输入或输出参数。

问题

只将简单类型映射到列。在 RPC/ 编码 Web 方法中使用的复杂类型或数组导致无法将 Web 方法映射到代理表。

用户操作

修改 Web 方法，仅使用简单类型和数组。

文档 / 文字结果被截断

问题

如果 Web 服务返回的数据多于全局变量 `@@textsize` 的值，则将会按照 `@@textsize` 指定的大小截断这些数据。因此，返回的数据可能不会形成有效的 XML 文档。

用户操作

在使用 `select` 命令截断文本或图像数据时，因为这是 Adaptive Server Enterprise 的预期行为，所以不会产生任何错误。但是，系统会为 ASE Web 服务引擎记录一条警告，因此需要检查 `consumer.log` 文件。

启动 ASE Web 服务引擎

- 问题** `runws` 脚本没有成功地启动 ASE Web 服务引擎。
- 用户操作**
- 1 确保尝试使用的端口未被其它进程使用。
 - 2 确保安装了正确的 JRE。ASE Web 服务需要 JRE 1.4.2 版或更高版本。
若要检查 JRE 版本，请在命令提示符处输入以下命令：

```
java -version
```
 - 3 如果要使用 `ws.properties` 以外的其它属性文件运行 Web 服务，则必须指定该文件的绝对路径。例如，要使用其它属性文件运行 ASE Web 服务引擎，请输入：

```
C:\sybase\WS-15_0\bin\runws -f
C:\sybase\WS-15_0\props\myfile.properties
```
- 问题** ASE Web 服务引擎在 `interfaces` 文件中找到了指定的 `ase_service_name`，但是 `producer.log` 却显示以下错误消息：
- ```
INFO [main] - Error locating libtcl.cfg file.
INFO [main] - java.io.FileNotFoundException: LDAP
config File does not exist
```

## 定位 WSDL

- 问题** 通过 Web 浏览器连接到 ASE Web 服务引擎的客户端无法找到 WSDL 文件，或者 ASE Web 服务引擎无法找到 WSDL 文件来执行代理表映射。
- 用户操作** 检验 ASE Web 服务引擎是否正在运行。如果通过浏览器使用 ASE Web 服务引擎，则应确保浏览器 URL 指明 `https://`（对于 SSL 连接）和 `http://`（对于标准连接）。

## 指定 ws.properties 中的条目

- 问题** 由于反斜杠 (\) 符号已用作转义字符，因此，使用单反斜杠符号的条目不会得到正确的解释。例如：
- ```
com.sybase.ase.ws.interfaces = d:\sybase\ini\sql.ini
```
- 用户操作** 用另一个反斜杠转义该反斜杠：
- ```
com.sybase.ase.ws.interfaces = d:\\sybase\\ini\\sql.ini
```
- 您也可以使用正斜杠：
- ```
com.sybase.ase.ws.interfaces = d:/sybase/ini/sql.ini
```

Windows NT 命令行参数

问题 当参数和参数值之间没有空格时，脚本将不会在 Windows NT 上运行。例如，按以下方式调用 *configssl* 脚本将不会执行该脚本：

```
configssl -dhostname
```

用户操作 在参数和相应的参数值之间放置一个空格：

```
configssl -d hostname
```

运行或停止脚本失败

问题 执行 *runws* 或 *stopws* 脚本失败。

用户操作 如果执行上述任一脚本失败，请执行以下操作：

- 检验 Adaptive Server Enterprise 是否正在运行。
- 确保指定的用户名和口令有效，可以登录到 Adaptive Server Enterprise。
- 检查 *producer.log* 或 *consumer.log* 文件中是否有任何错误消息。
- 检验是否可以在 *libtcl.cfg* 文件指向的 LDAP 服务器或 Adaptive Server Enterprise 的 *interfaces* 文件中找到所提供的 *ase_service_name*。

注释 在 Windows 系统上，*interfaces* 文件的名称为 *sql.ini*。

- 检验用户名是否具有 *sa_role* 特权。

空口令

问题 Adaptive Server Enterprise 用户的口令可以设置为空字符串。

用户操作 在符合以下条件的地方使用标识 *nopasswordspecified*：任何需要口令的地方，包括 *runws* 和 *stopws* 脚本。

指定使用 SSL 的 SOAP 结束点

- 问题** ASE Web 服务方法或示例应用程序没有返回带有在调用时指定的 *aseServerName* 的结果。
- 用户操作** 确保 *aseServerName* 名称是有效的 SOAP 结束点。如果使用的是 DNS 别名，请确保该别名可以解析为有效的 SOAP 结束点。如果正在使用 SSL，请确保 *aseServerName* 指定的结束点的名称与使用 *configssl* 脚本创建 SSL 认证时所提供的名称相同。例如：

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

此处，在调用 ASE Web 服务方法或示例应用程序时提供的 *aseServerName* 的值必须是 “https://mydomainhostname:8182”。如果在使用 SSL 时用 “localhost” 或 IP 地址来替代 *aseServerName*，则该方法或示例应用程序将不会返回结果。

sp_webservices ‘add’ 异常终止

- 问题** 与按 Ctrl+C 中断或 Adaptive Server Enterprise 崩溃时相同，在 *sp_webservices* 异常终止后，将保留执行 *sp_webservices* ‘add’ 选项的过程中所创建的代理表。
- 用户操作** 如果 *sp_webservices* 异常终止，请使用 *remove* 选项删除由 *add* 选项创建的所有代理表。

Web 服务代理表限制

- 问题** 如果某个代理表与通过执行 *sp_webservices* 命令的 *add* 选项而生成的 Web 服务代理表名称相同，则该代理表可能会被错误地指定为 Web 服务代理表。对 Web 服务代理表连接所进行的处理与对常规代理表连接所进行的处理是不同的。
- 用户操作** 将 Web 服务代理表限制在唯一数据库上。
- 问题** 在涉及 Web 服务代理表的查询中，*where* 子句不支持保留字 *or*。这对于在内部被转换为使用保留字 *or* 的查询，可能会产生错误。例如，以下查询从 Web 服务代理表 *testint* 中进行选择：

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

该查询在内部被转换为以下查询：

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

由于内部转换使用了保留字 *or*，因此用户提交的查询会产生错误。

用户操作

改变查询，使其内部转换不使用保留字 `or`。例如，使用临时表可以更改上述用户提交的查询：

```
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 表条目

问题

执行 `sp_webservices 'add'` 时，会发生以下错误：

```
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.
```

用户操作

`sysattributes` 表中必须有对应于 Adaptive Server Enterprise 的条目。要提供一个条目，请使用 `sp_addserver` 存储过程：

```
sp_addserver ase_entry, local, ase_entry
```

其中，`ase_entry` 是 `interfaces` 或 `sql.ini` 文件中的 Adaptive Server Enterprise 的本地名。然后，重新启动 Adaptive Server Enterprise。

诊断工具

ASE Web 服务提供诊断的详细记录和 JDBC 级跟踪，以帮助确定连接方面的问题。

详细记录

要为诊断启用详细记录，请相应地修改 *props* 目录中的 *logging.properties* 文件，然后重新启动 ASE Web 服务引擎。

以下是 *logging.properties* 文件（随 ASE Web 服务一起提供）的内容的示例：

```
# Set logging level for ASE Web Services Consumer
log4j.logger.com.sybase.ase.ws.sds=INFO, C
#log4j.logger.com.sybase.ase.ws.sds=DEBUG, C
log4j.additivity.com.sybase.ase.ws.sds=false

# Set logging level for ASE Web Services Producer
log4j.logger.com.sybase.ase.ws.producer=INFO, P
#log4j.logger.com.sybase.ase.ws.producer=TRACE, P
log4j.additivity.com.sybase.ase.ws.producer=false
```

有关详细的诊断记录，请按照以下方式修改 *logging.properties*：

```
# Set logging level for ASE Web Services Consumer
#log4j.logger.com.sybase.ase.ws.sds=INFO, C
log4j.logger.com.sybase.ase.ws.sds=DEBUG, C
log4j.additivity.com.sybase.ase.ws.sds=false

# Set logging level for ASE Web Services Producer
#log4j.logger.com.sybase.ase.ws.producer=INFO, P
log4j.logger.com.sybase.ase.ws.producer=TRACE, P
log4j.additivity.com.sybase.ase.ws.producer=false
```

启用 JDBC 级跟踪

要启用 JDBC 级跟踪，请参考相应的 jConnect 文档。

消息

表 6-1 列出了 ASE Web 服务的消息。

表 6-1: ASE Web 服务消息

消息号	消息文本
15200	没有 Web 方法映射到 WSDL URI [%s] 的代理表。
15201	大多数情况下, WSDL URI 的后缀为 ?WSDL。请检验 WSDL URI。
15202	由于数据类型不受支持, Web 方法 [%s] 没有映射到代理表。
15203	收到执行未知过程 [%s] 的请求。
15204	捕获到 IOException。这通常表示在 ASE 和 ASE Web 服务引擎之间的通信中发生了错误。 \n 详细信息: [%s]。
15205	捕获到 SQLException。这通常表示从 ASE 中检索元数据时发生了错误。 \n 详细信息: [%s]。
15206	捕获到未知的异常: 详细信息: [%s]。
15207	捕获到远程 Web 方法例外 (AxisFault)。这表示在远程 Web 方法中存在错误。 \n 详细信息: [%s]。
15208	捕获到映射例外。这表示在将 Web 方法参数映射到 ASE 类型时发生了错误。 \n 详细信息: [%s]。
15209	捕获到服务例外。这通常表示 WSDL 文件不正确。 \n 详细信息: [%s]。
15210	收到了格式不正确的对 Web 方法的 XML 输入。
15211	调用 Web 方法 (MalformedURLException) 时发生错误。详细信息: [%s]。
15212	捕获到 RemoteException。这通常表示在网络传输中发生了错误。 \n 详细信息: [%s]。
15213	调用 Web 方法时发生错误 (未知异常)。详细信息: [%s]。
15214	正在中止调用 Web 方法 [%s] (从代理表 [%s]), 因为 Web 方法预期有 [%s] 个参数, 收到了 [%s] 个。
15215	参数计数 / 类型不匹配。请检查传递给内置函数 ws_admin 的参数的数目和类型。
15220	使用该名称的用户定义的 Web 服务已存在。
15221	当服务正在引用别名时, 无法删除该别名。
15216	向内置函数 ws_admin 指定了未知操作 '%.*s'。请检查参数的拼写和位置。
15217	从 sysattributes 更新 / 插入 / 删除期间发生失败。
15218	无法检索当前数据库名。
15219	无法检索 '%.*s' 的对象 ID。
19307	正在使用 sds [%!1] 为 WSDL URI: [%2!] 生成代理表。
19308	为 [%!1] 找到了 WSDL 匹配。
19309	在系统中未找到 WSDL URI [%!1]。

消息号	消息文本
19310	正在用 [%2!] 更新 WSDL URI [%1!] 的超时条目。
19311	正在删除所有的 Web 服务元数据
19312	正在删除 WSDL URI [%1!] 的条目。
19313	要删除 Web 服务，必须提供一个有效的 SDS 服务器。未找到 SDS 服务器 [%1!]。请使用 sp_addserver 添加 SDS 服务器。
19319	要添加 Web 服务，必须提供一个有效的 SDS 服务器。\\n 未找到 SDS 服务器 [%1!]。\\n 请使用 sp_addserver 添加 SDS 服务器。
19320	检验 ASE Web 服务引擎是否正在运行。
19321	要添加 web 服务，必须指定一个有效的 WSDL URI。
19322	系统中已存在指定的 WSDL URI [%1!]。\\n 请首先使用 sp_webservices 将其删除。
19323	要列出关于系统中未加载的 web 服务的信息，必须提供一个 SDS 服务器。未找到 SDS 服务器 [%1!]。\\n 请使用 sp_addserver 添加 SDS 服务器。
19324	必须指定要修改的特定 wsdl uri。
19325	必须为修改操作指定要更改什么。
19326	[%1!] 对于 sp_webservices 修改不是有效的选项。
19327	必须为修改操作指定 item=value 语法。
19408	无法删除别名，因为它根本不存在。
19409	请指定要添加的别名的名称。
19410	请指定与此别名相关联的数据库名。
19412	无法重命名配置的服务。

有关 sp_webservices 的帮助信息，请在 isql 命令行中输入 sp_webservices help。

附录

本部分包含本手册的所有附录。

安装内容

本附录介绍 ASE Web 服务安装的内容。

ASE Web 服务目录树

ASE Web 服务与 Adaptive Server Enterprise 的根目录安装在同一级别。ASE Web 服务根目录的名称为 *WS-15_0*，它包含以下子目录：

表 A-1: ASE Web 服务目录

目录名	内容
<i>bin</i>	用于配置和运行 ASE Web 服务的脚本。
<i>lib</i>	ASE Web 服务所使用的 Java 库和 Java 软件包。
<i>logs</i>	日志文件的缺省位置。
<i>producer</i>	ASE Web 服务引擎在运行时的文件和子目录。
<i>props</i>	ASE Web 服务属性的文件。
<i>samples</i>	用 ASE Web 服务引擎建立并运行示例客户端的示例脚本。

bin 目录内容

下表介绍 *bin* 目录的内容。

表 A-2: bin 目录内容

文件 / 目录名	作用
<i>configssl</i>	配置 SSL。
<i>getpass.exe</i>	用于 Adaptive Server Enterprise 登录。此文件仅在 Windows 上可用。
<i>installws</i>	安装 <i>sp_webservices</i> 存储过程。
<i>runtcpmon</i>	创建一个监控器以跟踪 SOAP 消息。
<i>runws</i>	启动 ASE Web 服务引擎。
<i>stopws</i>	停止 ASE Web 服务引擎。

注释 这些文件的 Windows 版本带 *.bat* 后缀。

lib 目录内容

下表介绍 *lib* 目录的内容。

表 A-3: lib 目录内容

文件 / 目录名	作用
<i>axis.jar</i>	Apache Axis 文件
<i>commons-discovery.jar</i>	Apache Axis 文件
<i>commons-logging.jar</i>	Apache Axis 文件
<i>javax.servlet.jar</i>	服务器小程序库
<i>jaxrpc.jar</i>	Apache Axis 文件
<i>log4j-1.2.4.jar</i>	log4j 记录程序
<i>mail.jar</i>	Apache Axis 文件
<i>org.mortbay.jetty.jar</i>	HTTP 服务器
<i>saaj.jar</i>	Apache Axis 文件
<i>sqlx.jar</i>	SQLX 文件
<i>tools.jar</i>	Java 工具（例如， <i>javac</i> 编译器）
<i>ws.jar</i>	ASE Web 服务引擎的代码
<i>ws_debug.jar</i>	ASE Web 服务引擎的调试程序代码
<i>wsdl4j.jar</i>	Apache Axis 文件
<i>xercesImpl.jar</i>	Xerces 分析程序
<i>xmlParserAPIs.jar</i>	Xerces 分析程序

logs 目录内容

下表介绍 *logs* 目录的内容。

表 A-4: logs 目录内容

文件 / 目录名	作用
<i>consumer.log</i>	记录由通过代理表调用远程 Web 服务的 Adaptive Server Enterprise 产生的信息和错误消息。
<i>http.log</i>	记录 Web 服务器活动。
<i>producer.log</i>	记录在 Adaptive Server Enterprise 接收到的 SOAP 请求所产生的所有信息和错误消息。

producer 目录内容

下表介绍 *producer* 目录的内容。

表 A-5: producer 目录内容

文件 / 目录名	作用
<i>index.html</i>	Web 服务浏览器接口的主页。
<i>keystore</i>	持有加密密钥。
<i>ui</i>	Web 服务浏览器接口的目录结构。
<i>WEB-INF</i>	Jetty 所需的目录结构。
<i>wscertificate.cer</i>	自动生成的 SSL 证书。仅当 <i>configssl</i> 脚本已经运行时，此文件才存在。

props 目录内容

下表介绍 *props* 目录的内容。

表 A-6: props 目录内容

目录名	作用
<i>logging.properties</i>	log4j 的配置文件
<i>ws.properties</i>	ASE Web 服务引擎的所有配置参数
<i>wsmmsg.properties</i>	消息的配置文件

samples 目录内容

samples 目录包含 Apache 和 Microsoft .NET 的预编译代码和源代码。您可以在自己的应用程序中使用源代码。下表介绍 *samples* 目录的内容。

表 A-7: samples 目录内容

目录名	作用
<i>apacheclient</i>	包含用于编译和运行示例客户端的示例脚本的目录
<i>ms.net</i>	包含 .NET 示例的目录

配置属性

本附录讨论以下内容：

主题	页码
ws.properties	77
myres.properties	79
指定属性文件条目	80

ws.properties

ws.properties 文件包含 ASE Web 服务的以下配置设置。

表 B-1: ws.properties 条目

<code>com.sybase.ase.ws.producer.httpport</code>	表示 ASE Web 服务引擎应该监听 HTTP 连接的端口。缺省值为 8181 。
<code>com.sybase.ase.ws.consumer.cisport</code>	表示 ASE Web 服务引擎应该监听 TDS 的端口。缺省值为 8183 。
<code>com.sybase.ase.ws.producer.logfilename</code>	表示 ASE Web 服务引擎的日志文件应存放的位置。对于 UNIX，缺省位置为 <code>\$\$SYBASE/WS-15_0/logs/producer.log</code> ；对于 Windows，缺省位置为 <code>%SYBASE%\WS-15_0\logs\producer.log</code> 。
<code>com.sybase.ase.ws.consumer.logfilename</code>	表示 ASE Web 服务引擎的日志文件应存放的位置。对于 UNIX，缺省位置为 <code>\$\$SYBASE/WS-15_0/logs/consumer.log</code> ；对于 Windows，缺省位置为 <code>%SYBASE%\WS-15_0\logs\consumer.log</code> 。
<code>com.sybase.ase.ws.producer.jettylogfile</code>	表示 HTTP 请求的日志文件应存放的位置。对于 UNIX，缺省位置为 <code>\$\$SYBASE/WS-15_0/logs/http.log</code> ；对于 Windows，缺省位置为 <code>%SYBASE%\WS-15_0\logs\http.log</code> 。
<code>com.sybase.ase.ws.interfaces</code>	表示 Adaptive Server Enterprise 的 <i>interfaces</i> 或 <i>sql.ini</i> 文件的位置。对于 UNIX，缺省位置为 <code>\$\$SYBASE/interfaces</code> ；对于 Windows，的缺省位置为 <code>%SYBASE%\ini\sql.ini</code> 。

`com.sybase.ase.ws.libtcl`

表示用于标识 LDAP 服务器的 `libtcl.cfg` 文件的位置。对于 32 位 UNIX 平台，缺省位置为 `$$SYBASE/$SYBASE_OCS/config/libtcl.cfg`；对于 32 位 Windows 平台，缺省位置为 `%SYBASE%\%SYBASE_OCS%\ini\libtcl.cfg`。对于 64 位 UNIX 平台，缺省位置为 `$$SYBASE/$SYBASE_OCS/config/libtcl64.cfg`；对于 64 位 Windows 平台，缺省位置为 `%SYBASE%\%SYBASE_OCS%\ini\libtcl64.cfg`。

`com.sybase.ase.ws.maxpostsize`

确定进入的 SOAP 请求的最大尺寸。缺省值为 20000。

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

确定是否激活了基于 Web 的用户界面。可以从 `https://hostname:https_port` 获得该用户界面。缺省值为 `true`。

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

表示服务 HTTP 端口的线程池中的最大线程数。缺省值为 250。

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

表示服务 HTTP 端口的线程池中的最大线程数。缺省值为 45。

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

表示线程可保持空闲状态的最长时间（以毫秒为单位）。缺省值为 60000。

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

表示服务 HTTPS 端口的线程池中的最大线程数。缺省值为 250。

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

表示服务 HTTPS 端口的线程池中的最大线程数。缺省值为 45。

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

表示线程可保持空闲状态的最长时间（以毫秒为单位）。缺省值为 60000。

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

表示 SSL 认证的口令。没有提供缺省值。

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

表示 SSL 的 keystore 的位置。对于 UNIX，缺省位置为 `$$SYBASE/WS-15_0/producer/keystore`；对于 Windows，缺省位置为 `%SYBASE%\WS-15_0\producer\keystore`。

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

表示 ASE Web 服务引擎应该监听 HTTPS 连接的端口。缺省值为 8182。

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

表示 SSL 的 keystore 口令。没有提供缺省值。

myres.properties

myres.properties 文件是在 Sybase Central 插件执行 ASE Web 服务的配置任务时创建的。*myres.properties* 文件包含 ASE Web 服务的以下配置设置。

表 B-2: myres.properties 条目

<i>Web 服务</i> — 请通过设置以下条目来配置 Web 服务。	
<code>ws.ini</code>	表示 Adaptive Server Enterprise 的 <i>interfaces</i> 或 <i>sql.ini</i> 文件的位置。对于 UNIX, 缺省位置为 <code>\$\$SYBASE/interfaces</code> ; 对于 Windows, 缺省位置为 <code>%SYBASE%\ini\sql.ini</code> 。
<code>ws.libtcl</code>	表示用于标识 LDAP 服务器的 <i>libtcl.cfg</i> 文件的位置。对于 32 位 UNIX 平台, 缺省位置为 <code>\$\$SYBASE/\$SYBASE_OCS/config/libtcl.cfg</code> ; 对于 32 位 Windows 平台, 缺省位置为 <code>%SYBASE%\%SYBASE_OCS%\ini\libtcl.cfg</code> 。对于 64 位 UNIX 平台, 缺省位置为 <code>\$\$SYBASE/\$SYBASE_OCS/config/libtcl64.cfg</code> ; 对于 64 位 Windows 平台, 缺省位置为 <code>%SYBASE%\%SYBASE_OCS%\ini\libtcl64.cfg</code> 。
<code>ws.producer.port</code>	表示 ASE Web 服务引擎的端口。
<code>ws.producer.log</code>	表示 <i>producer.log</i> 文件的位置。
<code>ws.producer.jettylogfile</code>	表示 HTTP 请求的日志文件应存放的位置。对于 UNIX, 缺省位置为 <code>\$\$SYBASE/WS-15_0/logs/http.log</code> ; 对于 Windows, 缺省位置为 <code>%SYBASE%\WS-15_0\logs\http.log</code> 。
<i>SSL</i> — 请通过设置以下条目来配置 SSL。	
<code>ws.ssl.host</code>	表示要访问的 SSL 主机的名称。
<code>ws.ssl.keystorelocation</code>	表示 SSL 的 keystore 的位置。对于 UNIX, 缺省位置为 <code>\$\$SYBASE/WS-15_0/producer/keystore</code> ; 对于 Windows, 缺省位置为 <code>%SYBASE%\WS-15_0\producer\keystore</code> 。
<code>ws.ssl.certificatepassword</code>	表示 SSL 认证的口令。没有提供缺省值。
<code>ws.ssl.keystorepassword</code>	表示 SSL 的 keystore 口令。没有提供缺省值。
<code>ws.consumer.name</code>	表示 <i>interfaces</i> 或 <i>sql.ini</i> 文件中指定的 ASE Web 服务引擎的名称。

`ws.consumer.host`

表示 ASE Web 服务引擎的主机。

`ws.consumer.port`

表示 ASE Web 服务引擎进程的端口号。

`ws.consumer.log`

表示 `consumer.log` 文件的位置。

installws — 请通过设置以下条目来运行 *installws* 脚本：

`ws.sqlsrv.server_name`

表示要运行 *installws* 的 Adaptive Server Enterprise 的名称。

`ws.sqlsrv.sa_login`

表示 Adaptive Server Enterprise 的用户登录名。

`ws.sqlsrv.sa_password`

表示 Adaptive Server Enterprise 的口令。

指定属性文件条目

由于反斜杠 “\” 符号已用作转义字符，因此，使用单反斜杠符号的条目不会得到正确的解释。例如：

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

要解决此问题，请再用一个反斜杠来转义该反斜杠。例如：

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

您也可以使用正斜杠。例如：

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

SOAP 和 Adaptive Server Enterprise 的数据类型映射

本附录说明 SOAP 和 Adaptive Server Enterprise 的数据类型。

数据类型映射

以下几节描述了 SOAP 与 Adaptive Server Enterprise 之间在 Web 服务功能方面的数据类型映射关系。

SOAP 到 ASE 数据类型映射

下表显示 SOAP 数据类型和 Adaptive Server Enterprise 中与之对应的类型。在将 RPC/ 编码 Web 服务映射到 Adaptive Server Enterprise 中的代理表时，将使用此表中的映射。

表 C-1: SOAP 和 Adaptive Server Enterprise 数据类型

SOAP 数据类型	Adaptive Server Enterprise 数据类型
string	varchar 长度取决于 Adaptive Server Enterprise 页的大小
boolean	smallint
float	real
double	double precision
decimal	float
duration	datetime
dateTime	datetime
time	datetime
date	datetime
gYearMonth	datetime
gYear	datetime
gMonthDay	datetime

SOAP 数据类型	Adaptive Server Enterprise 数据类型
gDay	datetime
gMonth	datetime
hexBinary	不支持
base64Binary	不支持
anyURI	varchar 长度取决于 Adaptive Server Enterprise 页的大小
QName	varchar 长度取决于 Adaptive Server Enterprise 页的大小
NOTATION	varchar 长度取决于 Adaptive Server Enterprise 页的大小
normalizedString	varchar 长度取决于 Adaptive Server Enterprise 页的大小
token	varchar 长度取决于 Adaptive Server Enterprise 页的大小
language	varchar 长度取决于 Adaptive Server Enterprise 页的大小
NMTOKEN	varchar 长度取决于 Adaptive Server Enterprise 页的大小
Name	varchar 长度取决于 Adaptive Server Enterprise 页的大小
NCName	varchar 长度取决于 Adaptive Server Enterprise 页的大小
ID	varchar 长度取决于 Adaptive Server Enterprise 页的大小
IDREF	varchar 长度取决于 Adaptive Server Enterprise 页的大小
ENTITY	varchar 长度取决于 Adaptive Server Enterprise 页的大小
integer	integer
nonPositiveInteger	integer
negativeInteger	integer
long	integer
int	integer
short	smallint
byte	tinyint
nonNegativeInteger	integer
unsignedLong	integer
unsignedInt	integer
unsignedShort	smallint

SOAP 数据类型	Adaptive Server Enterprise 数据类型
unsignedByte	tinyint
positiveInteger	integer
soap 数组	不支持
soap 复杂数据类型	不支持

create service 命令的 ASE 到 SOAP 数据类型映射

下表定义了可用作存储过程类型的数据类型与用于 SOAP 用户定义 Web 服务的数据类型之间的关系。

注释 对于通过 HTTP GET 方法（此方法对应于 http 或 raw 类型的 Web 服务）访问的 Web 服务，所有参数均映射到 xsd:string 类型。

表 C-2: SOAP 和 Adaptive Server Enterprise 数据类型

ASE 数据类型	SOAP 参数类型
tinyint	xsd:int
smallint	xsd:int
int	xsd:int
bigint	xsd:decimal
numeric (p,s)	xsd:decimal
decimal (p,s)	xsd:decimal
float (precision)	xsd:float
double (precision)	xsd:double
real	xsd:double
smallmoney	xsd:int
money	xsd:int
smalldatetime	xsd:datetime
datetime	xsd:datetime
date	xsd:datetime
time	xsd:datetime
char(n)	xsd:string
varchar(n)	xsd:string
unichar	xsd:string
univarchar	xsd:string
nchar	xsd:string
nvarchar	xsd:string
binary	xsd:byte[]
varbinary	xsd:byte[]
xml	xsd:string
text	不支持
image	不支持
Java 数据类型	不支持

词汇表

ASE Web 服务引擎	充当 Web 服务的生产者和使用者的 ASE Web 服务组件。ASE Web 服务引擎使客户端应用程序能够将 Adaptive Server Enterprise 存储过程和 SQL 作为 Web 方法来访问。ASE Web 服务引擎还将 Web 方法映射到代理表，以允许客户端应用程序通过 SQL <code>select</code> 语句调用 Web 方法。
DTD	文档类型定义，用于定义 XML 文档的合法构件块。DTD 可以在 XML 文档内部声明或从外部引用。
HTTP	超文本传输协议，是 Internet 协议套件的应用层的一部分，并且是 World Wide Web 中交换信息的主要方式。
HTTPS	采用 SSL 或 TLS 加密的 HTTP。
LDAP	轻量目录访问协议，是一个用于基于 Internet 的目录服务的应用程序级协议。
RPC/ 编码	一类通过 SOAP 消息调用的 Web 方法，其中每个方法参数在 SOAP 消息中都有一个 XML 元素。
SDS	专业数据存储，用作组件集成服务 (CIS) 来将 Web 方法映射到代理表中。
SOAP	简单对象访问协议，是一个通过网络传送基于 XML 的消息的标准。
SQLX	SQL/XML，一种 ANSI 和 ISO 标准，可以为在 SQL 数据库系统的上下文中使用 XML 提供支持。
UDDI	通用描述、发现和集成。
URI	统一资源标识符，是标识 Internet 资源的字符串。最常见的 URI 是统一资源定位符 (URL)，用于标识 Internet 地址。不太常见的 URI 是统一资源名 (URN)。
URL	统一资源定位符，URI 的一种类型。
W3C	World Wide Web 协会，该协会为 World Wide Web 制定软件标准。
Web 方法	WSDL 描述的功能，通过 SOAP 消息调用。

Web 服务	由 WSDL 文件描述的一个或多个 Web 方法。
WSDL	Web 服务描述语言，用于描述 Web 服务的公共接口。
Xerces	Apache 开放源代码 XML 分析程序。
XML	可扩展标记语言，一种经过 W3C 标准化的标记语言。
XML 模式	由结构和内容约束组成的 XML 文档的说明。
XPath	XML 路径语言，是一种用于为 XML 文档的各部分寻址的语言。
XQL	XML 查询语言，是 XQuery 的前身。
模式	用于定义 XML 文档的结构、内容及语义的大纲。
文档 / 文字	一类 Web 方法，通信方根据合并到 WSDL 文件中的 XML 模式指定要传输和格式化的数据。
用户定义的 Web 服务	由最终用户命名的、使用 SOAP 客户端或通过 Web 浏览器执行的任意 SQL 语句。

索引

英文

- add 选项
 - 参数 35
 - 用法 36
 - 语法 35
- addalias 选项
 - 参数 45
 - 用法 45
 - 语法 45
- Apache
 - 示例客户端 51
- ASE
 - 到 SOAP 的数据类型映射 81
- ASE Web 服务
 - 安全性 24
 - 方法 31
 - 目录树 73
 - 配置 21
 - 日志记录 49
 - 日志文件 49
 - 使用 29
 - 许可 23
 - 优点 4
- ASE Web 服务引擎
 - HTTP 处理程序 14
 - SOAP 处理程序 14
 - XML 映射程序 15
 - 代理表 18
 - 启动 29
 - 生产者 Web 方法 15
 - 生产者组件 14
 - 使用 29
 - 使用者组件 17
 - 用户定义的 Web 服务 15
 - 作为生产者 13
 - 作为使用者 16
- bin 目录 74
- configssl 24
- consumer.log 50
- create service 命令 41
 - 参数 41
 - 示例 42, 43, 44
 - 语法 41
- deploy 选项
 - 参数 46
 - 审计 48
 - 用法 46
 - 语法 45
- drop service 命令 44
 - 参数 44
 - 示例 44
 - 语法 44
- dropalias 选项
 - 参数 46
 - 用法 46
 - 语法 46
- execute 方法 32
 - 参数 32
 - 示例 33
 - 语法 32
- help 选项
 - 参数 36
 - 用法 36
 - 语法 36
- HTTP 处理程序 14
- http.log 50
- JDBC 级跟踪 67
- LDAP 5
- lib 目录 74
- list 选项
 - 参数 37
 - 用法 37
 - 语法 37

- listalias 选项
 - 用法 47
 - 语法 47
- listudws 选项
 - 参数 46
 - 用法 46
 - 语法 46
- log4j 50
- logging.properties 文件 23
- login 方法 34
 - 参数 34
 - 用法 34
 - 语法 34
- logout 方法 34
 - 用法 34
 - 语法 34
- logs 目录 75
- Microsoft .NET
 - 安装 SSL 证书 26
 - 示例客户端 55
- Microsoft .NET 的证书 26
- modify 选项
 - 参数 37
 - 语法 37
- myres.properties 文件
 - 内容 79
- producer 目录 75
- producer.log 50
- props 目录 75
- remove 选项
 - 参数 37
 - 语法 37
- RPC/ 编码 Web 方法 18
- runws 失败 64
- samples 目录 76
- SOAP 10
 - Web 服务标准 10
 - 到 ASE 的数据类型映射 81
 - 消息结构 11
- SOAP 处理程序 14
- SOAP 堆栈 18
- sp_webservices
 - add 选项 35
 - addalias 选项 45
 - deploy 选项 45
 - dropalias 选项 46
 - help 选项 36
 - list 选项 37
 - listalias 选项 47
 - listudws 选项 46
 - modify 选项 37
 - remove 选项 37
 - undeploy 选项 47
 - 使用 35
 - 与用户定义的 Web 服务 45
- SQL 4
- SSL
 - 配置 24
- stopws 失败 64
- Sybase Central 50
- undeploy 选项
 - 参数 47
 - 审计 49
 - 用法 47
 - 语法 47
- Web 方法
 - RPC/ 编码 18
 - 文档 / 文字 18
- Web 服务
 - 标准 5
 - 调用 38
 - 概述 3
- Web 服务描述语言, 请参见 WSDL 9
- ws.properties 文件 23
 - 内容 77
 - 条目 63
- WSDL 9
 - Web 服务标准 9
 - 定位 63
- wsmmsg.properties 文件 24
- XML 6
 - Web 服务标准 6
 - 文档结构 7
- XML 映射程序 15

A

- 安全性 5, 24
 - 用户定义的 Web 服务 47
- 安装
 - 内容 73

C

- 处理程序
 - HTTP 14
 - SOAP 14
- 存储过程 4

D

- 代理表 18
- 调用 Web 服务 38
 - 示例 38, 39

F

- 方法 31
 - execute 32
 - login 34
 - logout 34

G

- 故障排除 61
- 故障排除问题 61
 - runws 或 stopws 失败 64
 - sp_webservices 异常终止 65
 - sysattributes 表条目 66
 - Web 服务代理表限制 65
 - ws.properties 中的条目 63
 - 定位 WSDL 63
 - 空口令 64
 - 命令行参数 64
 - 启动 ASE Web 服务引擎 63
 - 未映射 RPC/ 编码 Web 方法 62

- 文档 / 文字结果被截断 62
- 远程服务器类定义设置 61
- 指定使用 SSL 的 SOAP 结束点 65

H

- 函数 4

J

- 简单对象访问协议, 请参见 SOAP 10

K

- 可扩展标记语言 6
- 空口令 64

M

- 目录
 - bin 74
 - lib 74
 - logs 75
 - producer 75
 - props 75
 - samples 76

P

- 配置 21
 - logging.properties 23
 - logging.properties 文件 23
 - ws.properties 23
 - ws.properties 文件 23
 - wsmmsg.properties 文件 24
 - 安装后 22
 - 安装期间 22
 - 属性 77
 - 文件 23

R

- 日志记录 49
 - consumer.log 50
 - http.log 50
 - producer.log 50
 - 设置策略 50
- 日志文件 49

S

审计

- deploy 选项 48
- undeploy 选项 49
- 用户定义的 Web 服务 48

使用 ASE Web 服务 29

示例应用程序 51

- Apache 客户端 51
- Execute.exe 56
- Microsoft .NET 客户端 55
- runexecute 52

属性

- com.sybase.ase.ws.consumer.cisport 77
- com.sybase.ase.ws.consumer.logfilename 77
- com.sybase.ase.ws.interfaces 77
- com.sybase.ase.ws.libtcl 78
- com.sybase.ase.ws.maxpostsize 78
- com.sybase.ase.ws.producer.httpport 77
- com.sybase.ase.ws.producer.jettylogfile 77
- com.sybase.ase.ws.producer.logfilename 77
- com.sybase.ase.ws.producer.ssl.httpport 78
- com.sybase.ase.ws.producer.ssl.keypassword 78
- com.sybase.ase.ws.producer.ssl.keystore 78
- com.sybase.ase.ws.producer.ssl.password 78
- com.sybase.ase.ws.producer.tuning.maxidletime 78
- com.sybase.ase.ws.producer.tuning.maxthreads 78
- com.sybase.ase.ws.producer.tuning.minthreads 78
- com.sybase.ase.ws.producer.tuning.ssl.maxidletime 78
- com.sybase.ase.ws.producer.tuning.ssl.maxthreads 78
- com.sybase.ase.ws.producer.tuning.ssl.minthreads 78
- com.sybase.ase.ws.ui.activate 78
- ws.consumer.host 80
- ws.consumer.log 80
- ws.consumer.name 79
- ws.consumer.port 80

- ws.ini 79
- ws.libtcl 79
- ws.producer.jettylogfile 79
- ws.producer.log 79
- ws.producer.port 79
- ws.sqlsrv.sa_login 80
- ws.sqlsrv.sa_password 80
- ws.sqlsrv.server_name 80
- ws.ssl.certificatepassword 79
- ws.ssl.host 79
- ws.ssl.keystorelocation 79
- ws.ssl.keystorepassword 79

属性文件

- 指定条目 80

数据类型

- create service 的 ASE 到 SOAP 映射 84
- SOAP 和 Adaptive Server Enterprise 81
- SOAP 与 ASE 间的映射 81
- 映射 81

W

- 文档 / 文字 Web 方法 18

X

- 详细记录 67
- 消息 68
- 许可 23

Y

- 用户定义的 Web 服务 5
 - 安全性 47
 - 命令 41
 - 审计 48
 - 使用 40
 - 与 sp_webservices 45

Z

- 诊断工具 66
 - JDBC 级跟踪 67
 - 详细记录 67

