

IBM InfoSphere DataStage
Version 11 Release 3

Web Services Pack Guide



IBM InfoSphere DataStage
Version 11 Release 3

Web Services Pack Guide



Note

Before using this information and the product that it supports, read the information in “Notices and trademarks” on page 57.

Contents

Chapter 1. Introduction 1

About Web Services technologies	2
Encoding requests and responses	2
Examples	2
Using the SOAP framework	3
Publishing Web Service operations	4
Accessing Web Services.	4
What is the Web Services Pack?	5
Web Services Transformer stage	5
Web Services Client stage	5
Web Service routines.	6
Implementing Web Service operations in server jobs	6

Chapter 2. Using the Web Service Meta

Data Importer 7

Supported standards.	7
Unsupported standards.	7
Supported files	7
About ASMX files	7
About DISCO files	7
Current limitations	8
Starting the Web Service Meta Data Importer	8
Accessing WSDL documents	9
Understanding the Web Service Explorer pane.	10
Importing Web Service operations	10
Using the automated import process	10
Using the XML Meta Data Importer	11
Understanding Web Service table definitions	12
Metadata table definitions	12
Mapper table definitions	12
Changing the home page of the Web Service Meta Data Importer	13

Chapter 3. Using the Web Services

Transformer stage 15

Required tasks in the Web Services Transformer stage	15
Importing a Web Service	15
Selecting a Web Service operation	16
Configuring a Web Service request	16
Configuring a Web Service response	17
Other tasks	18
Using SOAP headers	18
Setting up HTTP and HTTPS security	19
Setting up proxy server information	19
Passing data through from the input link to the output link	19
Processing SOAP faults	19
Setting a time-out factor for service requests	20
Defining a Reject link	20
Maintaining Web Service request information	20
Connecting stages to a server job	21
Setting up Stage properties	21
Setting up General properties	21

Setting up Options properties	22
Setting up Security properties	23
Setting up HTTP and HTTPS proxy properties	23
Setting up Input Link properties	23
Setting up General properties	24
Setting up Input Message properties	24
Setting up Input Header properties	25
Maintaining Columns properties	25
Setting up Output Link properties.	26
Setting up General properties	26
Setting up Output Message properties	26
Setting up Output Header properties	27
Maintaining Columns properties	27

Chapter 4. Using the Web Services

Client stage 29

Using a Web Service as a data source.	29
Examples	29
Using a Web Service as a data target	29
Examples	30
Getting started with configuring the Web Services Client stage	30
Required tasks in Web Services Client stage	31
Importing a web service	31
Selecting a web service operation	31
Configuring a Web Service request	31
Configuring a Web Service response	32
Optional tasks	33
Using SOAP headers	33
Setting up HTTP and HTTPS security	33
Setting up HTTP and HTTPS proxy server information	34
Using a time-out factor	34
Maintaining Web Service request information	34
Processing SOAP faults	34
Using reference links to supply input values	35
Connecting stages to a server job	35
Setting up Stage properties	35
Setting up General properties	36
Setting up Options properties	37
Setting up Security properties	37
Setting up HTTP and HTTPS proxy properties	37
Setting up Input Link properties	38
Setting up General properties	38
Setting up Input Message properties	38
Setting up Input Header properties	39
Maintaining Columns properties	40
Setting up Output Link properties.	40
Setting up General properties	40
Setting up Input Argument properties	40
Setting up Output Message properties	42
Setting up Output Header properties	44
Maintaining Columns properties	44

Chapter 5. Creating Web Service routines	45
About input arguments	45
Unsupported input processing	45
About return values	45
Unsupported output processing	45
Required tasks in Web Service routines	46
Importing a Web Service	46
Selecting a Web Service operation	46
Examining the Web Service routine	46
Testing the Web Service routine.	47
Appendix A. Product accessibility	49
Appendix B. Contacting IBM	51
Appendix C. Accessing the product documentation	53
Appendix D. Providing feedback on the product documentation	55
Notices and trademarks	57
Index	63

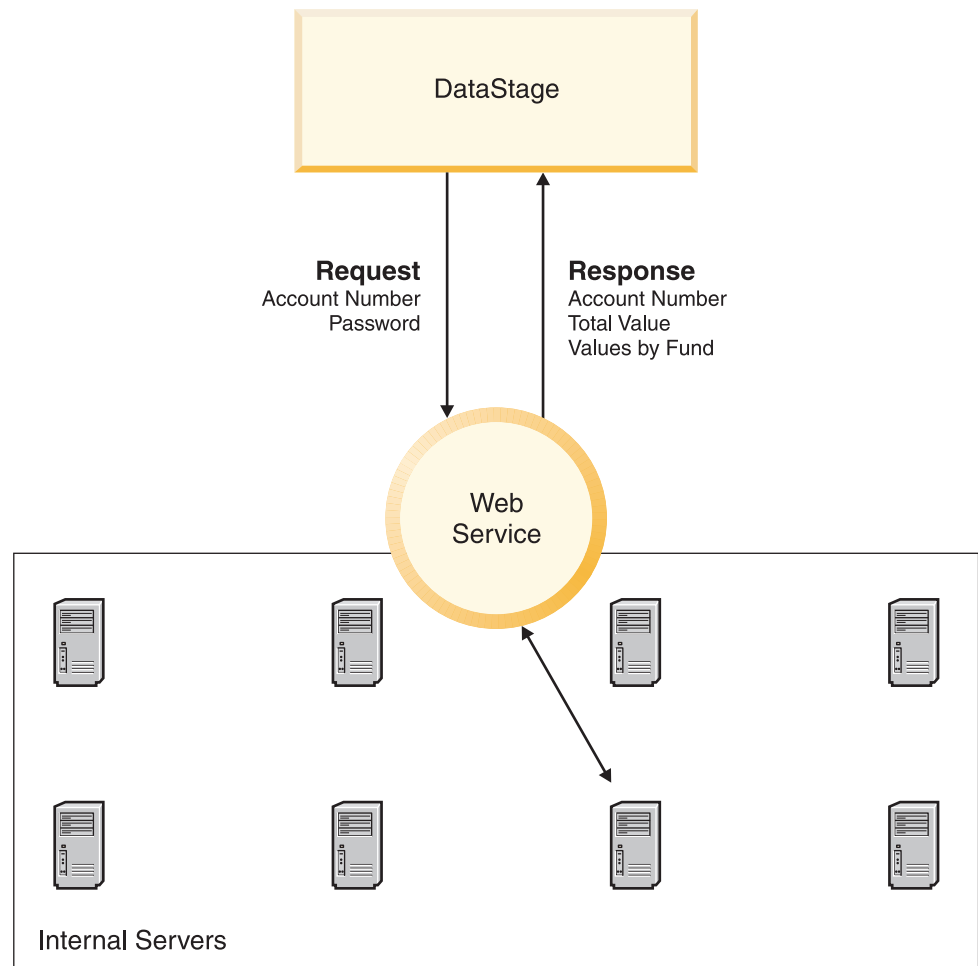
Chapter 1. Introduction

A web service describes:

- An application that is published and made available through a network, such as the internet.
- Recent and well-established protocols and standards that make access to the application possible.

A web service exposes one or more operations, which are functions that you call from another application, such as the InfoSphere® DataStage® and QualityStage® Designer. The operation accepts a request and returns a response. For example, an electronic brokerage may deploy a web service that returns portfolio information when you submit an account and password.

The following diagram illustrates the flow in a transaction between a web service and InfoSphere DataStage. The web service is a self-contained application, which may interface with other applications within a company.



About Web Services technologies

The protocols and standards needed to invoke a web service are represented in the following technology stack. The roles of each layer are discussed in the sections that follow.



Encoding requests and responses

Simple Object Access Control (SOAP) is a messaging framework for exchanging information between applications and web services. When SOAP is used, web service requests and responses are encoded in Extensible Markup Language (XML), a well-established text-based standard. XML elements and attributes identify the data that is exchanged between the web service and IBM® InfoSphere DataStage. Using an XML schema, the web service stipulates which data is needed for the exchange.

Examples

Consider the following request and response. The requester application queries a web service with a company name, Samples Outdoor Company. The web service returns an address block.

Sample request

```
<?xml version="1.0"?>
  <Address>
    <getAddress>
      <name>Samples Outdoor Company</name>
    </getAddress>
  </Address>
```

Sample response

```
<xml version="1.0"?>
  <Address>
    <getAddressResponse>
      <number>50</number>
      <street>Washington</street>
      <city>Westborough</city>
      <state>MA</state>
      <zip>01581</zip>
    </getAddressResponse>
  </Address>
```

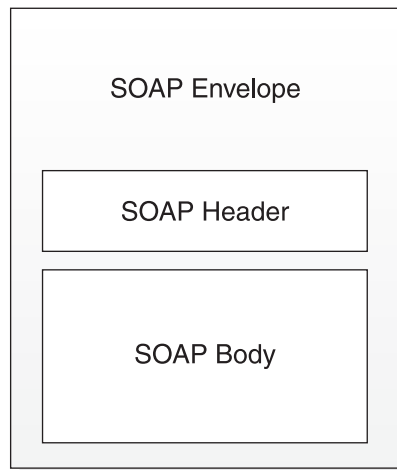

Using the SOAP framework

Published by the Worldwide Web Consortium (W3C), the SOAP specification describes the following items:

- Structure of SOAP messages, which includes the requests, responses, and other information.
- Rules for encoding data types as XML, from simple data types, such as strings and integers, to complex data types, such as classes and structures. In a SOAP message, the `encodingStyle` attribute identifies the URI that supplies the encoding rules.
- Conventions for invoking remote procedure calls (RPCs), from a requester application and a web service. Web Services Pack supports RPC-style and document-style communication with a web service.
- Binding of SOAP to transport protocols, such as HTTP and SMTP.

SOAP Message structure

A SOAP message has the following structure:



The SOAP envelope is a wrapper element that identifies the subordinate elements as a SOAP message and provides namespace declarations. Namespaces provide semantic context for elements within the SOAP body.

The SOAP header is an optional element that can contain metadata, such as authentication information, localization support, and delivery routes.

The SOAP body contains the payload of the message, which is either the web service request or web service response. The response can be a processing error, which is called a SOAP fault.

Requests that are incorporated within a SOAP message

Following is a sample request that is incorporated within a SOAP message. Major elements such as the `<SOAP-ENV:Body>` are highlighted.

```
<SOAP-ENV:Envelope  
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:auth="http://schemas.xmlsoap.org/ws/2002/04/secext"  
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">  
<SOAP-ENV:Header>  
  <auth:Security>
```

```

    <auth:UsernameToken>
      <auth:Username>Smith</auth:Username>
      <auth:Password>XMLER</auth:Password>
    </auth:UsernameToken>
  </auth:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  <ns:getAddress xmlns:ns="PhoneNumber">
    <name xsi:type="xsd:string"> Samples Outdoor Company </name>
  </ns:getAddress>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Publishing Web Service operations

Web services publish their operations and locations through a document written in an XML-based language called Web Services Description Language (WSDL).

WSDL documents specify:

- Operations that the web service offers. The collection of operations is called a port type.
- URL of the web service that IBM InfoSphere DataStage invokes. This information is represented by the service element.
- Structures for the web service request and response of each operation. These structures are called messages.
- Transport protocol for communication between a web service client and the web service. For example, SOAP over HTTP. This information is part of a binding.
- Mechanism for sending individual web service requests, such as RPCs. This is another part of a binding.

Accessing Web Services

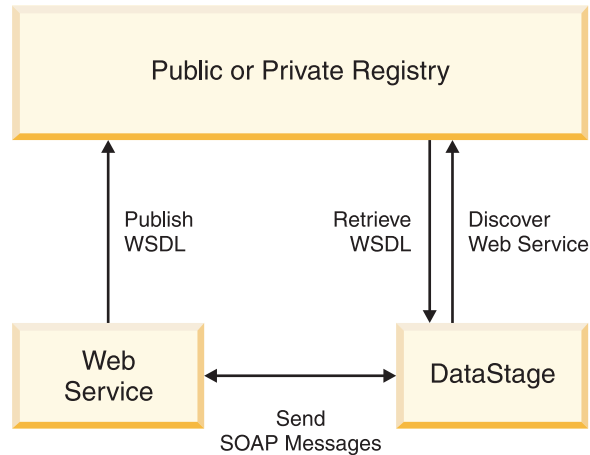
A growing number of companies that host web services publish their services through a registry. Some of these registries comply with the Universal Description, Discovery, and Integration (UDDI) specification. The UDDI specification sets standards for describing businesses that host web services and for finding or discovering them through a network connection. Typically, a UDDI registry has a web interface that contains direct or indirect links to WSDL documents.

A UDDI Business Registry (UBR) is a public subscriber service. Any company can enroll in a UBR, and anyone can search this information. Private registries are often made available only through password access. Using the Web Services Pack, you have direct access to several UBRs.

UBR data fall in these groups:

- White pages, which contain contact and general information about a company
- Yellow pages, which include taxonomies such as product categories and North American Industry Classification System (NAICS) identifiers
- Green pages, which contains technical specifications of the web service

The following diagram illustrates the relationship between IBM InfoSphere DataStage, a web service, and web service registries.

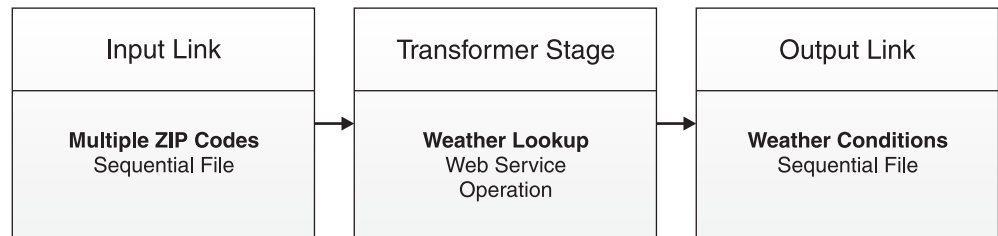


What is the Web Services Pack?

Using the Web Services Pack, you can access web service operations within an IBM InfoSphere DataStage Server job. Web Services Pack includes plug-in stages and added functionality in InfoSphere DataStage Server routines.

Web Services Transformer stage

Use this active stage when you need to use both an input link and an output link in a web service operation. In a simple weather query, the input stage provides a set of ZIP codes, and the output link stores corresponding weather conditions in a sequential file.



Web Services Client stage

Use this passive stage when you need the web service to act as either a data source or a data target during an operation. In addition, you do not need both input and output links in a single web service operation.

In a simple time query, the Client stage supplies a single Eastern Standard Time value and the output link stores the equivalent Greenwich Mean Time value.



Web Service routines

Using an InfoSphere DataStage Server routine, you can invoke a web service operation and use the response within the built-in Transformer stage.

Implementing Web Service operations in server jobs

About this task

Implementing web services in your server jobs using plug-in stages involves these major steps:

Procedure

1. Locate the web service using the Web Service Meta Data Importer.
2. Import one or more web service operations and its parameters into InfoSphere DataStage. This step creates table definitions based on WSDL information.
3. Add the appropriate Web Services Pack stage to your server canvas.
4. Select the web service operation that you want to call.

Chapter 2. Using the Web Service Meta Data Importer

You must import message and metadata information from the WSDL document for each web service operation that you want to call from a Web Services Pack stage and a web service routine. WSDL documents are parsed and converted to table definitions.

To access and import WSDL documents, use the Web Service Meta Data Importer.

Supported standards

The following standards and specifications are supported by the Web Service Meta Data Importer:

- SOAP 1.1 binding over HTTP
- Literal and SOAP-encoded web service arguments
- RPC-style and document-style operations

Unsupported standards

The following standards and specifications are not supported by the Web Service Meta Data Importer. In the plug-in stages, you can access web services that use these standards and specifications. However, you must manually create table definitions from their WSDL documents.

- SOAP 1.1 binding over non-HTTP transports
- SOAP 1.1 faults in web service responses
- SOAP 1.1 readers
- SOAP 1.2
- Web Services Inspection (WS-Inspection) documents for finding WSDL documents
- UDDI registry access through native resources such as API calls

Supported files

The Web Service Meta Data Importer can parse the following types of files:

- WSDL documents
- Microsoft ASP.NET Web Services (ASMX) files
- Discovery of Web Services (DISCO) files

About ASMX files

For ASMX files, the Importer displays the HTML content and finds the WSDL document that is associated with the DISCO file.

About DISCO files

For DISCO files, the Importer displays the DISCO contents in the **Web Browser** pane and a pop-up window containing the DISCO catalog from which you select a WSDL document.

Current limitations

The Web Service Meta Data Importer is not able to access a WSDL, DISCO, or ASMX file through HTTPS, through a proxy, or if any HTTP authentication is required. In those cases, use a regular Web Browser to access the WSDL file, save it locally on your workstation, and open this saved file from the Web Service Meta Data Importer.

Starting the Web Service Meta Data Importer

About this task

To start the Web Service Meta Data Importer:

Procedure

1. Open the InfoSphere DataStage Designer.
2. Select **Import > Table Definitions > Web Service Table Definitions**.

The **Web Service Meta Data Importer** opens.



Accessing WSDL documents

Depending on the web service, you access WSDL documents directly or through ASMX or DISCO files.

Accessing WSDL documents directly

About this task


Perform one of the following steps:

- Enter the URL for the WSDL document in the Address field.
- Search for the WSDL document through a web service directory link, such as XMethods.
- Open a WSDL document that resides on your file system or network.

The WSDL document opens in the **Web Browser** pane.

Opening a local or network file:

Procedure

1. Click the  icon.
The Open Local or UNC File dialog box opens.
2. Locate the WSDL document.
3. Click **Open**.

Accessing WSDL documents through a DISCO file

About this task

To access a WSDL document through a DISCO file:

Procedure

1. Perform one of the following steps:
 - Enter the URL for the DISCO file in the Address field.
 - Search for the DISCO file through a web service directory link, such as XMethods.The Importer displays the contents of the DISCO file in the **Web Browser** pane, and the DISCO catalog in an ASP.NET Discovery dialog box.
2. Double-click the web service whose WSDL document you want to parse.
The WSDL document itself or an HTML page containing a link to the document opens in the **Web Browser** pane. The **Web Service Explorer** pane contains a tree view of the web service port and operation.
3. If an HTML page opens, click the link for the WSDL document.

Accessing WSDL documents through an ASMX file

About this task

Perform one of the following steps:

- Enter the URL for the ASMX file in the Address field.
- Search for the ASMX file through a web service directory link, such as XMethods.
- Open an ASMX file that resides on your file system or network.

The Importer displays the readable description of the web service in the **Web Browser** pane and WSDL elements in the **Web Service Explorer** pane.

Understanding the Web Service Explorer pane

The **Web Service Explorer** pane displays a tree view of several WSDL elements when the Importer detects a WSDL document in the **Web Browser** pane.

Note: This pane does not expose all WSDL elements, such as <binding_name>.

Port type

The port type (SQL Terms and DefinitionsSoap) identifies the collection of operations that the web service offers. Port types are imported as metadata into a property table definition.

Input and output messages

The input message (GetTermsSoapIn) encodes a request to the web service as a SOAP message. The output message (GetTermsSoapOut) encodes the web service response. Both messages are imported as table definitions.

Importing Web Service operations

During an import, input and output messages and related metadata for a web service operation are mapped to columns of IBM InfoSphere DataStage table definitions. In a single step, you can import multiple operations.

You have two choices for creating the table definitions:

- Use the Web Service Meta Data Importer to automatically map the metadata and save table definitions. This approach is appropriate for most imports.
- Use the XML Meta Data Importer to adjust the automatic mappings and to save the table definitions. This is appropriate for advanced cases.

Using the automated import process

About this task

To map metadata automatically and save table definitions:

Procedure

1. Highlight one or more operations in the **Web Service Explorer** pane.
2. Right-click your selections, and select **Import**.
The Import Progress dialog box opens. Each input and output message corresponds to a separate task.
3. To see the list of imported messages, click **Details**.
If there are errors for any message, the Status column shows Error. For more information about errors, see Viewing Import Errors.
4. Click **Close** to complete the import.

In the InfoSphere DataStage repository browser, the web service name is used as a table definition category:

```
Table Definitions\WebServices\service_name
```


Viewing import errors

About this task

Import errors often indicate problems with parsing faulty WSDL documents.

From the Errors pane, you can:

- Copy errors to the clipboard.
- Access tracing information.
- Diagnose the errors using the XML Meta Data Importer.

To view errors, click the **Error** tab on the Errors pane.

Copying errors to clipboard:

About this task

To copy an error message to the Windows clipboard, right-click the message and select **Copy**.

Accessing tracing information:

About this task

To access tracing information, right-click a message and select **Details**.

Accessing XML Meta Data Importer:

About this task

To access the XML Meta Data Importer, right-click a message and select **XML Meta Data Importer**.

Using the XML Meta Data Importer

About this task

Using the XML Meta Data Importer, you process one operation at a time. You make decisions about the input message, followed by the output message.

To import an operation:

Procedure

1. Right-click an operation in the **Web Service Explorer** pane.
2. From the pop-up menu, select **Import using XML Meta Data Importer**.
The **XML Meta Data Importer** opens, with one or more nodes of the input message selected in the upper left pane.
3. Modify the selections, as needed.
4. When finished, select **File > Save**.
5. Select **File > Return to Web Service Meta Data Importer**.
The **XML Meta Data Importer** re-opens, with one or more nodes of the output message selected.
6. Modify the selections, as needed.
7. When finished, select **File > Save**.
8. To return to the **Web Service Meta Data Importer**, choose **File > Return to Web Service Meta Data Importer**.

Understanding Web Service table definitions

The table definitions that are created during an import fall into two groups that you can access through IBM InfoSphere DataStage:

- Metadata that you should not edit or rename
- Mapper information that you should not edit or rename

Metadata table definitions

Do not modify or rename the following table definitions:

Table definition	Contents
Info_WS	Web service metadata, including these properties: <ul style="list-style-type: none">• Name of the web service• Operations list• Port address for the protocols used (binding)• Port name• Web service URL
<i>operation name_OP</i>	Operation metadata, including these properties: <ul style="list-style-type: none">• SOAP action, such as URI• SOAP binding style, such as document
<i>operation name_MSGIN</i>	Input message metadata, including these properties: <ul style="list-style-type: none">• Body Use attribute: literal or encoded• Encoding style (serialization format)• Name• Namespace
<i>operation name_MSGOUT</i>	Output message metadata: <ul style="list-style-type: none">• Body Use attribute: literal or encoded• Encoding style (serialization format)• Name• Namespace

Mapper table definitions

Do not rename these table definitions if you want the stages to automatically load them.

Table definition	Contents
<i>operation_IN</i>	Input parameters for the web service operation, which are sent to the web service during a request.
<i>operation_OUT</i>	Output parameters that the web service sends to InfoSphere DataStage as a response.

Changing the home page of the Web Service Meta Data Importer

When you start the Importer, the file `defaultpage.htm` opens in the **Web Browser** pane. It is located in the InfoSphere DataStage Designer client directory. You can change the home page by editing the file `defaultpage.htm`.

Chapter 3. Using the Web Services Transformer stage

Use the Web Services Transformer stage to:

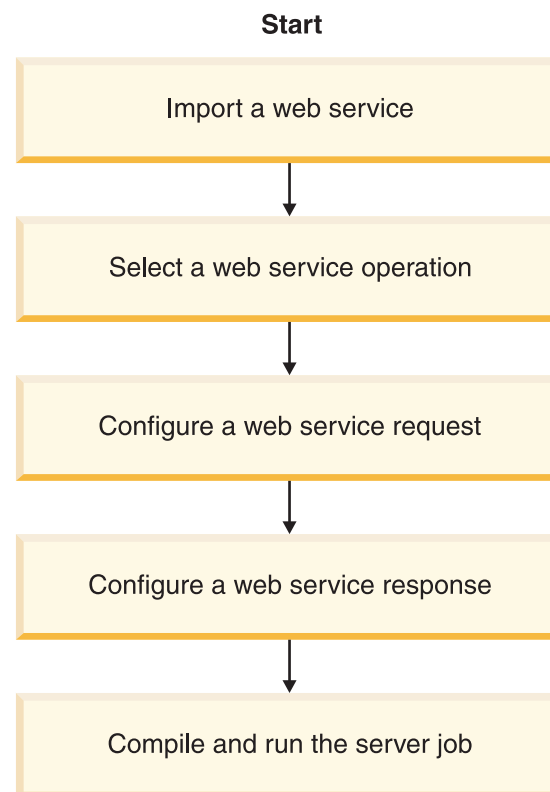
- Send a request to a web service operation using parameter values from an input link.
- Direct the web service response to an output link.
- Define a Reject link for requests that fail.

The Web Services Transformer stage encodes requests as SOAP messages and decodes responses from SOAP messages, using metadata that is defined for a web service operation. As an alternative, you can send the web response to another stage for decoding.

The following sections describe the concepts and tasks involved in setting up a Web Services Transformer stage. The tasks are divided into ones that are required for all web service operations and ones that pertain to specific requirements.

Required tasks in the Web Services Transformer stage

The following diagram illustrates the minimum tasks required to set up a Web Services Transformer stage.



Importing a Web Service

The first step in using web service operations is importing the web service. For more information, see [Using the Web Service Meta Data Importer](#).

Selecting a Web Service operation

A Web Services Transformer stage calls a single operation. To select the operation, you use the Web Service Browser, which you access through the Stage properties tab.

For more information about selecting a web service operation, see [Setting up Stage Properties](#).

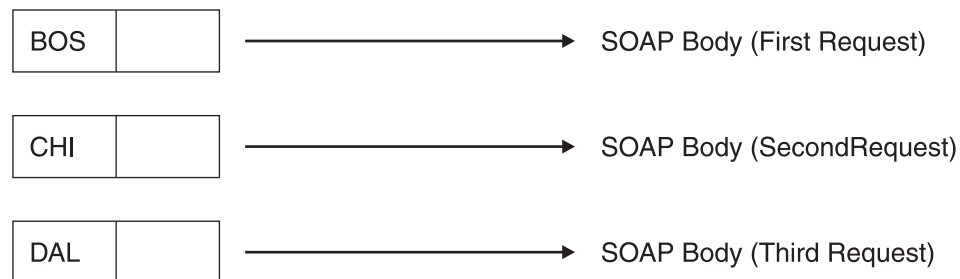
Configuring a Web Service request

The Web Services Transformer stage sends requests, one row at a time to the web service, which are encoded in a SOAP message. If the data required for the request is stored in one row, configure a request using the Web Services Transformer stage and the appropriate data extraction stage. This involves importing the operation metadata from the input message table definition as input link properties.

In the simplest case, a web service operation expects a single parameter value as input to an operation, and your input data contains multiple rows with a single value. Each row is processed in a separate request.

For example, you send a city name in a request, and the web service operation returns current temperature. The following diagram illustrates three input rows that supply city values for different requests against the same web service operation.

Input Rows



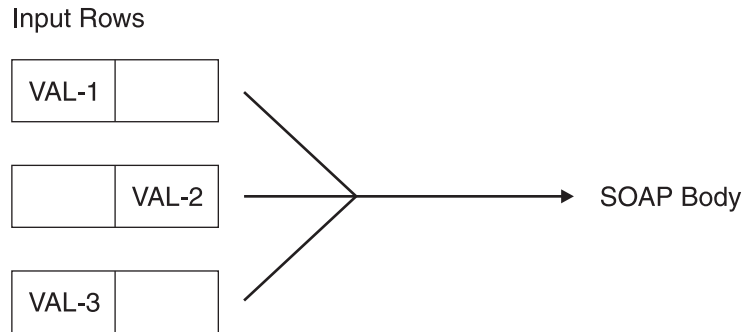
For more information about configuring these requests, see [Setting up Input Link Properties](#).

Handling arrays of input values

About this task

If the web service expects in one request an array of values, and these values are distributed across multiple data rows, use another stage such as XML Output to aggregate the values. The final output must be an XML document that complies with the web service WSDL. This XML document becomes the source for the SOAP message sent to the web service.

The following diagram illustrates three input rows that supply values for the same request.



**Example: Computing total sales:
About this task**

The web service operation computes total sales by product ID for a specific customer. There are two data sets:

- Product ID and quantity that you supply in a web service request
- Unit prices found in a database that the web service accesses

Here is sample input:

Product ID	Quantity	Customer ID
Z100	2	0444
Z142	15	0634
Z555	1	0646
Z432	35	0444

For this operation, you need to:

Procedure

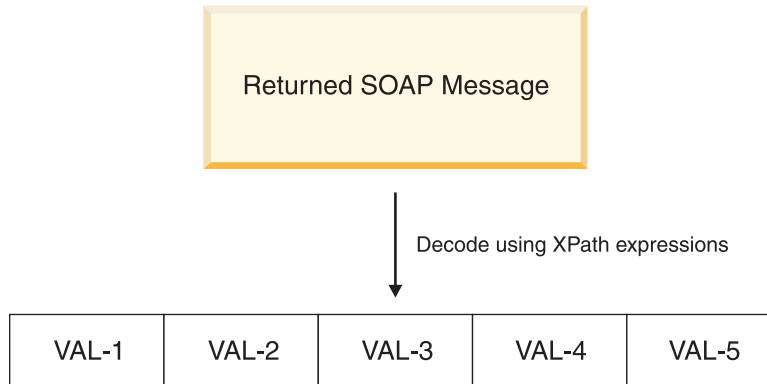
1. Filter the raw data based on customer ID; for example, customer **0444**. Use the appropriate database stage to filter the records.
2. Aggregate the multiple rows in the result set as an XML document. To do this, use the XML Output plug-in stage.
3. In the Web Services Transformer stage, identify the column that contains the XML document.

Results

For more information about configuring these requests, see [Setting up Input Link Properties](#).

Configuring a Web Service response

A web service returns a SOAP message that the Web Services Transformer stage can decode into columns of one output row using XPath expressions.



To configure this option, you create output link properties that import parameters and namespace data from an output message table definition.

As alternatives to decoding the response into a single row, you can:

- Decode the SOAP message into multiple rows using XML Input as a linked stage.
- Convert the content to another XML format using XML Transformer as a linked stage.
- Use the SOAP message without modification in a linked stage.
- Perform data transformations using the appropriate stages.

For more information about configuring a web service response, see [Setting up Output Link Properties](#).

Other tasks

This section describes tasks that are not required for all web service operations but may be required for your server jobs.

Using SOAP headers

A web service may use SOAP headers to exchange information with a requester application. Within a server job, an input column supplies the input header elements, which generate a `<SOAP-ENV:Header>` block at run time. If the web service returns an output header that you want to use in your server job, you can extract it to an output column.

Examples

The XYZ web service expects a user ID and password in the input header and returns a session ID in the output header. You need to supply an input header. Optionally, you can extract the output header.

The DEF web service expects user ID and password as input arguments and does not return session information. No SOAP headers are involved.

For information about SOAP header requirements, consult the web service WSDL.

For information about setting up an input header, see [Setting up Input Header Properties](#).

For information about extracting an output header, see [Setting up Output Header Properties](#).

Setting up HTTP and HTTPS security

If the HTTP server that hosts the web service uses Basic HTTP authentication, you must supply a user ID and password with the web service request.

If the HTTP server uses an HTTPS connection, you have two choices for authenticating the server:

- Trust the server implicitly.
- Trust the server if its public key certificate is stored in a local keystore file.

For more information about setting up credentials and policies, see [Setting up Security Properties](#).

Setting up proxy server information

If you need to pass web service requests through a proxy server, you must identify the server and its port. In addition, you may need to supply a user ID and password.

For more information about setting up proxy server information, see [Setting up HTTP and HTTPS Proxy Properties](#).

Passing data through from the input link to the output link

The output link supports a passthrough mechanism by which data is copied without modification from the input link to the output link. This mechanism works with columns that are not involved in the web service operation. It requires an exact match between column names specified in both the output link and the input link.

For example, you can copy comments from an input table column to an output table. If the Comments column is not part of the web service request but is included in the input and output links, the Comments value is automatically copied. You can disable the passthrough mechanism.

Note: When a string column is a passthrough column, the corresponding output link column must also be defined as a string column (and can not be defined as a `ustring` column).

The WSTransformer stage has a `propagate` column in order to automatically copy input link column data to output link column data.

For information about disabling the passthrough mechanism, see [Setting up Output Link Properties](#).

Processing SOAP faults

SOAP faults contain error messages returned by a web service. The Web Services Transformer stage does not parse SOAP faults. However, you can log SOAP faults. The following table describes the processing options.

Processing option	Result
Reject	Rejected rows are sent to a Reject link, if one exists. If there is no Reject link, the row is ignored, and a warning message containing the SOAP fault is logged. You can disable logging the warning message by selecting the Disable Log Reject Reasons option.
Fatal	The web service operation terminates when a SOAP fault is returned.
Warning	The SOAP fault is logged as a warning.
Info	The SOAP fault is logged as an information message
Trace	The SOAP fault is logged only if you activate tracing before running the server job. Activate tracing in the InfoSphere DataStage Director. To set up SOAP fault handling, see Setting up Stage Properties.

Setting a time-out factor for service requests

Using a time-out factor, you can specify the maximum processing time for a web service request. By default, the Web Services Transformer stage imposes no limit. However, the web service may set one.

To set a time-out factor, see Setting up Stage Properties.

Defining a Reject link

You can set up a Reject link that receives:

- Input rows, including data sent to the web service and other columns
- SOAP faults

A Reject link receives data only when the web service returns a SOAP fault. For information about setting up a Reject link, see Setting up Output Link Properties.

Maintaining Web Service request information

As an alternative to using the Web Service Browser, you can manually enter the web service information. In addition, you can modify this information after using the Web Service Browser.

You can edit a limited number of parameters like the web service name (Service Name) without affecting access to a web service. However, when you change most parameters such as port address (Port Address) and SOAP action (SOAP Action), you send a different request or access a different web service.

WSDL Element / URL	Web Service Information Label	Guideline
<service name>	Service Name	Description
<operation name>	Operation Name	Affects access

WSDL Element / URL	Web Service Information Label	Guideline
<address location>	Port Address	Affects access
<port name>	Port Name	Description
(WSDL URL)	WSDL Address	Description
<operation soapAction>	SOAP Action	Affects access
<binding transport=... style=>	Operation Style	Affects access
xmlns	Input Message Namespace	Affects access

For more information about editing web service request information, see [Setting up Stage Properties](#).

Connecting stages to a server job

About this task

Using the InfoSphere DataStage Designer, add a Web Services Transformer stage to your server job diagram.

Procedure

1. From the Real Time category of the Palette pane, drag the Web Services Transformer stage icon onto the canvas.
2. Click the Link icon, and connect two stages in your diagram.
3. Repeat Step 2 as needed.

Setting up Stage properties

The Stage properties are divided among four pages:

- General
- Options
- Security
- Proxy

Setting up General properties

About this task

Use the General page to:

- Select a web service operation.
- Edit web service information.
- Add a general description of the operation.

Selecting a web service operation

Procedure

1. In the **Web Services Transformer** stage dialog box, click the **General** tab.
2. Click **Select Web Service Operation**.

The **Web Service Browser** window opens.

3. In the **Web Services** pane, highlight the service name.
You must import the web service if it is not listed. For more information, see “Starting the Web Service Meta Data Importer.”
4. In the **Operations** pane, highlight the operation that you want to use in the Web Services Transformer stage.
5. Click the **Select this item** link in the **Information** pane.
The **Web Services Transformer stage** dialog box reappears. The web service name and the selected operation name appear in the General page.

Accessing the imported metadata Procedure

1. On the General page, click **Advanced**.
The Web service information dialog box opens.
2. Modify values as needed, using the guidelines presented in Maintaining Web Service Request Information.
3. Click **OK** to save the values and return to the General page.

Adding a general description of the web service operation About this task

In the **Description** box, optionally enter a description about the web service operation.

Starting the Web Service Meta Data Importer About this task

If the web service that you want to select an operation from is not listed in the **Web Services** pane, you can start the Web Service Meta Data Importer without exiting from the Web Service Browser.

To start the Web Service Meta Data Importer, click the  icon.

Setting up Options properties About this task

Use the Options page to

- Set up processing options for SOAP faults.
- Set a time-out limit for web service responses.

Procedure

1. Click the **Options** tab. When the web service returns SOAP faults, the default action is terminate the server job. This action corresponds to the Fatal option.
2. If needed, select a different error handling option, as described in Processing SOAP Faults.
3. To prevent sending a warning message to the job log about a rejected web service request, select the **Reject** option and the **Disable Log Reject Reasons** check box.
4. To set a time-out factor for the web operation, enter a maximum, in units of seconds. The value zero (0) is equivalent to no limit.

Note: The web service may impose a time limit that is shorter than your time-out factor.

Setting up Security properties

About this task

Use the Security page to:

- Add user and password credentials for authentication against the HTTP server that hosts the web service.
- Set up trust criteria for HTTPS connections.

Procedure

1. Click the **Security** tab.
2. For Basic HTTP authentication, select **Authorization Required** and enter valid user name and password credentials.
3. For SSL encryption, select **SSL Encryption Required**. Then:
 - To accept security credentials of all HTTPS servers, select **Trust All Servers**.
 - To accept security credentials of only HTTPS servers whose information is stored in a keystore file, enter a local keystore file path. Web Services Pack accepts keystore files created by the JavaSoft JDK keytool utility.

Obtain the server certificate from your HTTP server administrator and create a keystore file using keytool. Import the server certificate in this keystore. For additional information, refer to the `-import` option of the keytool utility in the JDK documentation.

Setting up HTTP and HTTPS proxy properties

About this task

Use the Proxy page to:

- Identify the HTTP or HTTPS proxy server.
- Identify your credentials that are required to access the internet.

Procedure

1. Click the **Proxy** tab.
2. Select **HTTP/HTTPS Proxy Required**.
3. If required by the proxy server, specify a user name and password.
4. Specify the proxy host name or IP address.
5. Specify the listening port on the proxy server.

Setting up Input Link properties

The Input Link properties are divided among four pages:

- General
- Input Message
- Input Header
- Columns

Input values may include one of the following items:

- One or more columns of input parameters that are part of the web service request.
- A column containing the SOAP message that was encoded by another stage that constitutes the web service request.

Input values may also include columns whose values are not sent to the web service but are copied to the output link.

Setting up General properties

About this task

Use the General page to record a description of the input link.

Procedure

1. Click the **General** tab, if needed.
2. Optionally, enter a general description of the input link.

Setting up Input Message properties

About this task

Input message properties define the contents of the SOAP message for a web service request. Use the Input Message page to perform one of these actions:

- Generate the SOAP message using information supplied by the WSDL of the web service.
- Load the namespace, input parameters, and other table definition information for the web service that you specify on the General page of the Stage properties page. This information is used to create the SOAP message for a web service request.
- Use the SOAP message that a previous stage generates.

Generating the SOAP message from the WSDL of the Web Service

About this task

First use the Web Service Meta Data Importer to import the operation.

Procedure

1. Click the **Input Message** tab.
2. To load namespace information and input parameters for the web service operation listed as a Stage property, click **Load Message Information**.

One of the following conditions applies:

- If you used the Web Services Meta Data Importer to import the web services, the generated table definition created is automatically selected and loaded. A window opens if there is conflict with an existing column.
- If you did not use the Web Services Meta Data Importer to import the web service, see “Loading input table definitions.”

Loading input table definitions

About this task

If you did not use the Web Services Meta Data Importer to import a web service, the Table Definitions dialog box opens.

The system highlights the table definition for the input message (*operation_IN*) of the operation that is listed on the **General** tab of the **Stage** properties page.

To use the *operation_IN* table definition:

Procedure

1. In the Table Definitions dialog box, accept the default table definition (*operation_IN*) by clicking **OK**.
The Select Columns dialog box opens.
2. Accept the column selections by clicking **OK**. The namespace information is populated, and the selected input columns are created.

Using the SOAP message from a previous stage About this task

To use a SOAP message that was generated by a previous stage:

Procedure

1. Click the **Input Message** tab.
2. Select the **User-Defined Message** check box.
3. In the **Choose the Column Receiving the User Message** list, select the column that contains the SOAP message that a previous stage supplies to the Web Services Transformer stage.

Note: If you configure the Web Services Transformer stage to handle user-defined messages or headers, you must define the column containing the message as VarBinary, especially if the message is not UTF-8 encoded.

Setting up Input Header properties About this task

Use the Input Header page to specify an input link column that supplies the SOAP header that is sent with the web service request.

Procedure

1. Click the **Input Header** tab.
2. Select the **User-Defined Header** check box.
3. In the **Choose the Column Receiving the User Header** list, select the column that contains the input header that a previous stage supplies to the Web Services Transformer stage.

Note: If you configure the WSTransformer stage to handle user-defined messages or headers, you must define the column containing the message as VarBinary, especially if the message is not UTF-8 encoded.

Maintaining Columns properties About this task

Use the Columns page to:

- Inspect the definitions of input values.
- Load another table definition. This is an advanced task.

Procedure

1. Click the **Columns** tab.

The input parameters supplied by the web service or the specified column from a linked stage appear in rows.

The **Description** property contains an XPath expression only when the input message is parsed and mapped.

2. Modify the information, as needed. For example, you can add columns whose values pass through to the output link.

Setting up Output Link properties

The Output Link properties are divided among four pages:

- General
- Output Message
- Output Header
- Columns

Output values can include one of the following items:

- One or more output parameters that are part of the web response.
- Column from a linked stage that receives the web response as is.

Setting up General properties

About this task

Use the General page to:

- Record a description of the output link.
- Identify the output link as a Reject link and the column that will contain the SOAP fault.
- Disable passthrough copying from the input link to the output link.

Procedure

1. Click the **General** tab, if needed.
2. Optionally, enter a general description of the output link.

Setting up Output Message properties

About this task

Use the Output Message page to perform one of these actions:

- Load namespace information and output parameters from the table definition that contains WSDL information. The Web Services Transformer stage uses this information to decode the web response into an output row.
- Specify a single column on the input link of the XML Input stage or another stage that receives the response without decoding from the web service.

Decoding output message from the Web Service

Procedure

1. Click the **Output Message** tab.
2. Click **Load Message Information**.

One of the following conditions applies:

- If you used the Web Services Meta Data Importer to import the web services, the generated table definition created is automatically selected and loaded. A window opens if there is conflict with an existing column.
- If you did not use the Web Services Meta Data Importer to import the web service, see “Loading output table definitions” on page 27.

Loading output table definitions

About this task

If you did not use the Web Services Meta Data Importer to import a web service, the Table Definitions dialog box opens.

The system highlights the table definition for the output message (*operation_OUT*) of the operation that is listed on the **General** tab of the **Stage** properties page.

To use the *operation_OUT* table definition:

Procedure

1. In the Table Definitions dialog box, accept the default table definition (*operation_OUT*) by clicking **OK**.
The Select Columns dialog box opens.
2. Accept the column selections by clicking **OK**.
The namespace information is populated, and the selected output columns are created.

Returning output messages without decoding

About this task

Procedure

1. Define a column in the input link of the receiving stage.
2. On the Output page of the Web Services Transformer stage, click the **Output Message** tab.
3. Select the **User-Defined Message** check box.
4. In the **Choose the Column Receiving the User Message** list, select the column of the linked stage that will receive the output message. This is the column that you defined in step 1.

Note: If you configure the Web Services Transformer stage stage to handle user-defined messages or headers, you must define the column containing the message as VarBinary, especially if the message is not UTF-8 encoded.

Setting up Output Header properties

About this task

Use the Output Header page to specify an output column that receives the output header that is sent in a web service response.

Procedure

1. Click the **Output Header** tab.
2. Select the **User-Defined Header** check box.
3. In the **Choose the Column Receiving the User Header** list, select the output column that receives the output header.

Maintaining Columns properties

About this task

Use the Columns page to:

- Inspect the definitions of output values.

- Load another table definition. This is an advanced task.

The Columns page can also include values that are passed from the input link.

Procedure

1. Click the **Columns** tab.

The output parameters supplied by the web service WSDL or the specified column from the linked stage appear in rows.

The **Description** property contains an XPath expression only when the output message is parsed and mapped.

2. Modify the information, as needed. For example, you can add columns whose values pass from the input link to the output link.

Chapter 4. Using the Web Services Client stage

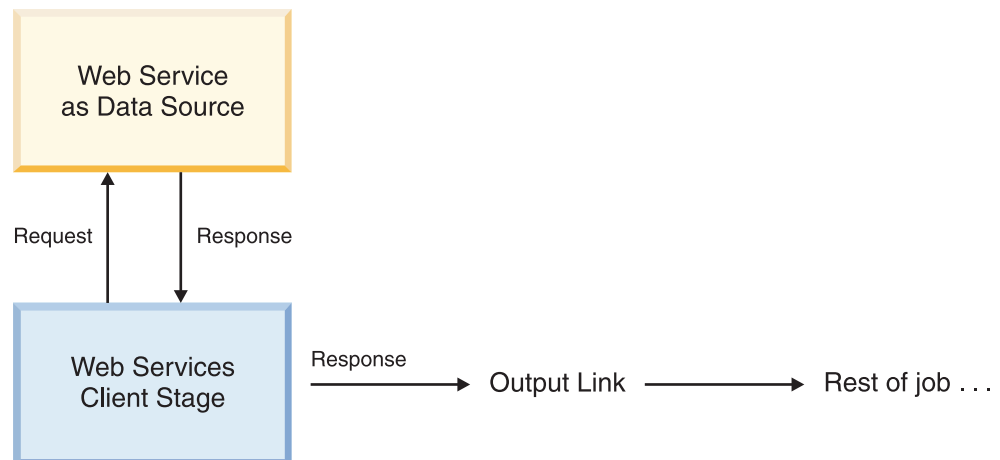
Use the Web Services Client stage to set up and call web service operations when:

- You need the web service to act as either a data source or a data target during an operation.
- You do not need both input and output links in a single web service operation.

The Web Services Client stage encodes requests as SOAP messages and decodes responses from SOAP messages, using metadata that is defined for a web service operation in its WSDL.

Using a Web Service as a data source

When the web service acts as a data source, the response returns to the Web Services Client stage and is sent to an output link.



When the web service is a data source, the input arguments are constants or job parameters that are created in the Web Services Client stage.

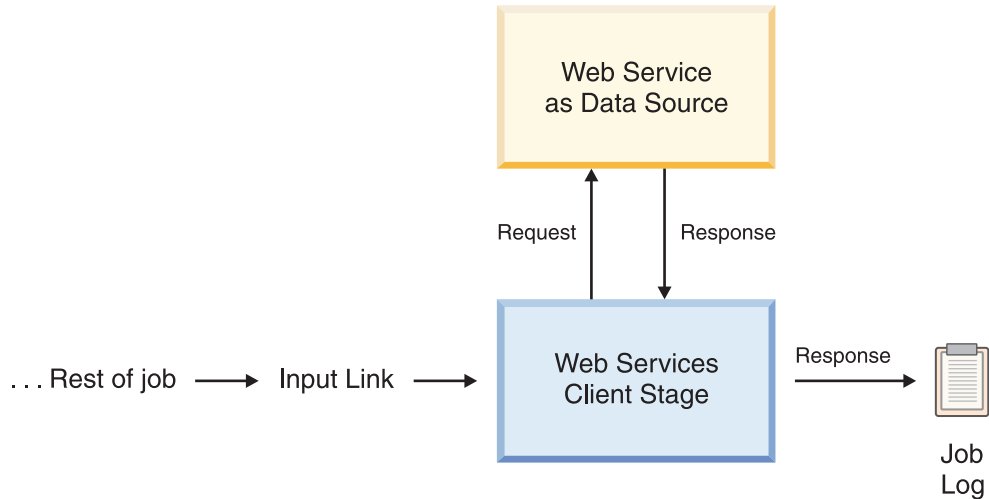
Examples

The following examples illustrate using a web service as a data source:

- Obtain purchase orders from a business partner for invoice processing.
- Acquire product listings and price changes from a retailer.
- Access daily prices and ratios from a stock brokerage.
- Obtain employee records from a Human Resources Management System.

Using a Web Service as a data target

When the web service acts as a data target, the response is typically an acknowledgement, which returns to the Web Services Client stage. You can log the response but cannot send it to an output link. If you need to send the response to an output link, use the Web Services Transformer Stage.



Examples

The following examples illustrate using a web service as a data target:

- Trigger a data restore from a backup server.
- Post part levels to a just-in-time Inventory Control module.
- Send orders for durable goods from a retailer to a vendor.
- Send monthly sales data to Corporate Sales.

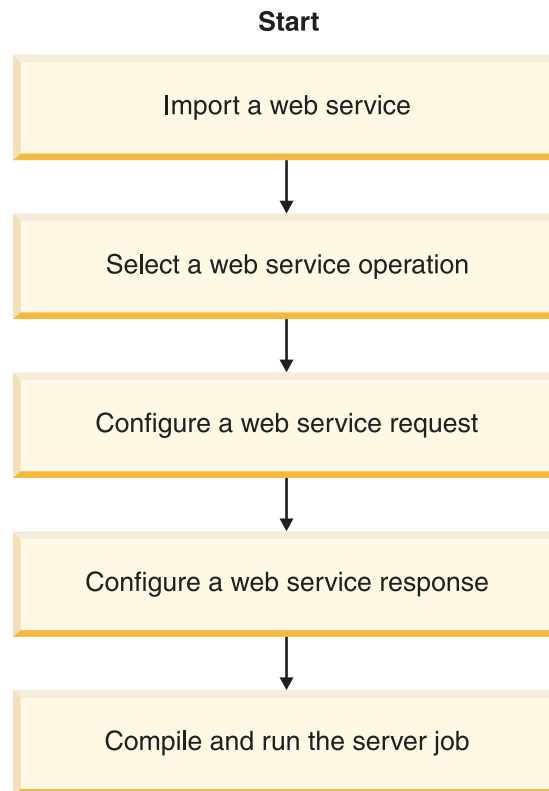
Getting started with configuring the Web Services Client stage

Setting up a server job that calls a web service depends partly on whether the invoked web service acts a data target or data source. For example, when the web service acts a data target, you do not configure the Web Services Client stage with an output link. In contrast, an output link and an output stage are needed when the web service acts as a data source.

The following sections describes concepts and tasks in required and optional categories. Differences in configuration that are based on web service roles are described. Each section refers you to procedures for configuring the Web Services Client stage.

Required tasks in Web Services Client stage

The following diagram illustrates the minimum tasks required to set up a Web Services Client stage.



Importing a web service

The first step in using web service operations is importing the web service. For more information, see [Using the Web Service Meta Data Importer](#).

Selecting a web service operation

A Web Service Client stage calls a single operation. To select the operation, you use the Web Service Browser, which you access through the Stage properties tab.

For more information about selecting an operation, see [Setting up General Properties](#).

Configuring a Web Service request

About this task

The Web Services Client stage sends requests one row at a time to the web service, which are encoded in a SOAP message. To configure a web service request, you must import metadata about the web service operation and supply input parameter values.

Web Service as data source

About this task

Supply constants or job parameters for input parameters, if needed, within the Web Services Client stage.

For more information about importing metadata and setting up input parameters, see [Setting up Input Argument Properties](#).

Web Service as data target

About this task

Set up an input link that provides the parameter values.

If the data required for the request is stored in one row, configure a request using the Web Services Client stage and, if needed, the appropriate data extraction stage.

If the web service expects in one request an array of values, and these values are distributed across multiple data rows, use another stage such as XML Output to aggregate the values.

For more information about:

- Data scenarios, see [Configuring a Web Service Request](#).
- Importing metadata and setting up an input link, see [Setting up Input Link Properties](#).

Configuring a Web Service response

Configuring a web service response varies by scenario.

Web Service as data source

A web service returns a SOAP message that the Web Services Client stage can decode into columns of one output row using XPath expressions. If you need to decode the SOAP message into multiple rows, use the XML Input stage.

Within the Web Services Client stage, import the metadata for the output message and decide how to process the web response.

For more information about:

- Data scenarios, see [Configuring a Web Service Response](#).
- Importing the metadata and processing the web response, see [Setting up Output Link Properties](#).

Web Service as data target

You can direct web service responses to a job log but not to an output link. There are three logging choices.

Logging option	Description
Never	Web service responses are never logged, regardless of the job trace setting. This is the default option.
Trace	Information entries that contain web service responses are logged only when job tracing is active. Activate tracing in the InfoSphere DataStage Director.

Logging option	Description
Info	Information entries that contain web service responses are logged, regardless of the job trace setting.

For information about setting logging options, see [Setting up General Properties](#).

Optional tasks

This section describes optional tasks. Tasks that apply to a specific scenario are noted.

Using SOAP headers

A web service may use SOAP headers to exchange information with a requester application. Within a server job, you can supply input header elements that are sent to the web service as a `<SOAP-ENV:Header>` block.

If the web service returns an output header that you want to use in your server job, you can extract it to an output column.

Examples

The XYZ web service expects a user ID and password in the input header and returns a session ID in the output header. You need to supply an input header. Optionally, you can extract the output header.

The DEF web service expects user ID and password as input arguments and does not return session information. No SOAP headers are involved.

For information about SOAP header requirements, consult the web service WSDL.

Setting up header processing

About this task

To specify an input header:

- When the web service acts as a data target, use the [Input Header](#) page, as described in [Setting up Input Header Properties](#).
- When the web service acts as a data source, use the [Input Arguments](#) page, as described in [Supplying an Input SOAP Header](#).

For information about extracting an output header, see [Setting up Output Header Properties](#).

Setting up HTTP and HTTPS security

If the HTTP server that hosts the web service uses Basic HTTP authentication, you must supply a user ID and password with the web service request.

If the HTTP server uses an HTTPS connection, you have two choices for authenticating the server:

- Trust the server implicitly.
- Trust the server if its public key certificate is stored in a local keystore file.

For more information about setting up credentials and policies, see [Setting up Security Properties](#).

Setting up HTTP and HTTPS proxy server information

If you need to pass web service requests through a proxy server, you must identify the server and its port. In addition, you may need to supply a user ID and password.

For more information about setting up proxy server information, see [Setting up HTTP and HTTPS Proxy Properties](#).

Using a time-out factor

Using a time-out factor, you can specify the maximum processing time for a web service request. By default, the Web Services Client stage imposes no limit. However, the web service may set one.

For information about setting a time-out factor, see [Setting up Options Properties](#).

Maintaining Web Service request information

As an alternative to using the Web Service Browser, you can manually enter the web service information. In addition, you can modify this information after using the Web Service Browser.

You can edit a limited number of parameters like the web service name (Service Name) without affecting access to a web service. However, when you change most parameters such as port address (Port Address) and SOAP action (SOAP Action), you send a different request or access a different web service.

WSDL element / URL	Web Service information label	Guideline
<service name>	Service Name	Description
<operation name>	Operation Name	Affects access
<address location>	Port Address	Affects access
<port name>	Port Name	Description
(WSDL URL)	WSDL Address	Description
<operation soapAction>	SOAP Action	Affects access
<binding transport=... style=>	Operation Style	Affects access
xmlns	Input Message Namespace	Affects access

For more information about editing web service request information, see [Setting up General Properties](#).

Processing SOAP faults

SOAP faults contain error messages returned by a web service. The Web Services Client stage does not parse SOAP faults. However, you can log SOAP faults.

The following table describes the processing options.

Processing option	Result
Reject	Rejected rows are sent to a Reject link, if one exists. If there is no Reject link, the row is ignored, and a warning message containing the SOAP fault is logged. You can disable logging the warning message by selecting the Disable Log Reject Reasons option.
Fatal	The web service operation terminates when a SOAP fault is returned.
Warning	The SOAP fault is logged as a warning.
Info	The SOAP fault is logged as an information message.
Trace	The SOAP fault is logged only if you activate tracing before running the server job. Activate tracing in the InfoSphere DataStage Director.

To set up SOAP fault handling, see [Setting up Options Properties](#).

Using reference links to supply input values

Application: Web services that are used as data sources.

Use a reference link when an external source supplies one or more input parameters for a web service request. To pass the parameters to the Web Services Client stage, use a built-in stage, such as the Transformer stage.

For more information about using a reference link, see [Setting up a Reference Link](#).

Connecting stages to a server job

About this task

Using the InfoSphere DataStage Designer, add a Web Services Client stage to your server job diagram.

Procedure

1. From the Real Time category of the Palette pane, drag the Web Services Client stage icon onto the canvas.
2. Connect other stages, based on whether the web service acts as a data source or data target.

If the web service acts as a data source, perform one of the following steps:

- Use an output stage with a stream link.
- Use a Transformer stage with a reference link.

If the web service acts as a data target, connect an input stage.

Setting up Stage properties

The Stage properties are divided among four pages:

- General
- Options
- Security

- Proxy

Setting up General properties

About this task

Use the General page to:

- Select a web service operation.
- Access a dialog box for editing web service information.
- Create a general description of the operation.

Selecting a web service operation

Procedure

1. In the **Web Services Client** stage dialog box, click the **General** tab.
2. Click **Select Web Service Operation**.
The **Web Service Browser** window opens.
3. In the **Web Services** pane, highlight the service name.
You must import the web service if it is not listed. For more information, see “Starting the Web Service Meta Data Importer.”
4. In the **Operations** pane, highlight the operation that you want to use in the Web Services Client stage.
5. Click the **Select this item** link in the **Information** pane.
The **Web Services Client stage** dialog box reappears. The web service name and the selected operation name appear in the General page.

Accessing the imported metadata

Procedure

1. On the General page, click **Advanced**.
The Web service information dialog box opens.
2. Modify values as needed, using the guidelines presented in Maintaining Web Service Request Information.
3. Click **OK** to save the values and return to the General page.

Adding a general description of the web service operation

About this task

In the **Description** box, optionally enter a description about the web service operation.

Starting the Web Service Meta Data Importer

About this task

If the web service that you want to select an operation from is not listed in the **Web Services** pane, you can start the Web Service Meta Data Importer without exiting from the Web Service Browser.

To start the Web Service Meta Data Importer, click the  icon.

Setting up Options properties

About this task

Use the Options page to:

- Set up processing options for SOAP faults.
- Set a time-out limit for web service responses.

Procedure

1. Click the **Options** tab.

When the web service returns SOAP faults, the default action is terminate the server job. This action corresponds to the **Fatal** option.

2. If needed, select a different error handling option, as described in Processing SOAP Faults.
3. To set a time-out factor for the web operation, enter a maximum, in units of seconds.

The value zero (0) is equivalent to no limit. The web service may impose a time limit that is shorter than your time-out factor.

Setting up Security properties

About this task

Use the Security page to:

- Add user and password credentials for authentication against the HTTP server that hosts the web service.
- Set up trust criteria for HTTPS connections.

Procedure

1. Click the **Security** tab.
2. For Basic HTTP authentication, select **Authorization Required** and enter valid user name and password credentials.
3. For SSL encryption, select **SSL Encryption Required**. Then:

- To accept security credentials of all HTTPS servers, select **Trust All Servers**.
- To accept security credentials of only HTTPS servers whose information is stored in a keystore file, enter a local keystore file path. Web Services Pack accepts keystore files created by the JavaSoft JDK keytool utility.

Obtain the server certificate from your HTTP server administrator and create a keystore file using keytool. Import the server certificate in this keystore.

For additional information, refer to the `-import` option of the keytool utility in the JDK documentation.

Setting up HTTP and HTTPS proxy properties

About this task

Use the Proxy page to identify the HTTP or HTTPS proxy server and your credentials that are required to access the internet.

Procedure

1. Click the **Proxy** tab.
2. Select **HTTP/HTTPS Proxy Required**.
3. If required by the proxy server, specify a user name and password.

4. Specify the proxy host name or IP address.
5. Specify the listening port on the proxy server.

Setting up Input Link properties

About this task

Application: Web services that are used as data targets.

The Input Link properties are divided among four pages:

- General
- Input Message
- Input Header
- Columns

Input values can include one of the following items:

- One or more columns of input parameters that are part of the web service request
- A column containing the SOAP message that was encoded by another stage that constitutes the web service request

Setting up General properties

About this task

Use the General page to:

- Make a decision about logging web service responses.
- Record a description of the input link.

Procedure

1. Click the **General** tab, if needed.
2. Using the Log Response options, select a logging option for web responses. For information about these options, see *Web Service as Data Target*.
3. Optionally, enter a general description of the input link.

Setting up Input Message properties

Input message properties define the contents of the SOAP message for a web service request. You can supply the SOAP message in two ways:

- Generate the SOAP message within this stage using information supplied by the web service WSDL.
- Use the SOAP message that a previous stage generates.

Note: In both scenarios, an input stage supplies the input values.

Generating the SOAP message from the WSDL of the Web Service

Procedure

1. Click the **Input Message** tab.
2. To load namespace information and input parameters for the web service operation listed as a Stage property, click **Load Message Information**.
One of the following conditions applies:

- If you used the Web Services Meta Data Importer to import the web services, the generated table definition is automatically selected and loaded. A window opens if there is conflict with an existing column.
- If you did not use the Web Services Meta Data Importer to import the web service, see “Loading input table definitions.”

Loading input table definitions:

About this task

If you did not use the Web Services Meta Data Importer to import a web service, the Table Definitions dialog box opens.

The system highlights the table definition for the input message (*operation_IN*) of the operation that is listed on the **General** tab of the **Stage** properties page.

To use the *operation_IN* table definition:

Procedure

1. Accept the default table definition (*operation_IN*) selection by clicking **OK**.
The Select Columns dialog box opens.
2. Accept the column selections by clicking **OK**.
The namespace information is populated, and the selected input columns are created.

Using the SOAP message from a previous stage

About this task

To use a SOAP message that was generated by a previous stage:

Procedure

1. Click the **Input Message** tab.
2. Select the **User-Defined Message** check box.
3. In the **Choose the Column Receiving the User Message** list, select the column that contains the aggregated XML that another stage supplies to the Client stage.

Note: If you configure the Web Services Client stage or Web Services Transformer stage to handle user-defined messages or headers, you must define the column containing the message as VarBinary, especially if the message is not UTF-8 encoded.

Setting up Input Header properties

About this task

Use the Input Header page to specify an input link column that supplies the SOAP header that is sent with the web service request.

Procedure

1. Click the **Input Header** tab.
2. Select the **User-Defined Header** check box.
3. In the **Choose the Column Receiving the User Header** list, select the column that contains the input header that a previous stage supplies to the Web Services Client stage.

Note: If you configure the Web Services Client stage to handle user-defined messages or headers, you must define the column containing the message as `VarBinary`, especially if the message is not UTF-8 encoded.

Maintaining Columns properties

About this task

Use the Columns page to:

- Inspect the definitions of input values.
- Load another table definition. This is an advanced task.

Procedure

1. Click the **Columns** tab.

The input parameters supplied by the web service or the specified column from a linked stage appear in rows.

2. Modify the information, as needed.

Setting up Output Link properties

Application: Web services that are used as data sources.

The Output Link properties are divided among five pages:

- General
- Input Arguments
- Output Message
- Output Header
- Columns

Output values can include one of the following items:

- One or more output parameters that are part of the web response
- A column from a linked stage that receives the web response as is

Setting up General properties

About this task

Use the General page to record a description of the output link.

Procedure

1. Click the **General** tab, if needed.
2. Optionally, enter a general description of the output link.

Setting up Input Argument properties

Use the Input Arguments page to set up input arguments under two scenarios: the output link is a stream link, or the output link is a reference link.

Stream link options

- Load the namespace, input parameters, and other table definition information for the web service that you specify on the General page of the **Stage** properties page. This information is used to create the SOAP message for a web service request.

- Specify constants or job parameters (*#param#*) for input parameters.

Reference link options

- Specify which input arguments are supplied by a reference link.

Common options for stream and reference links

- Supply input SOAP header elements.

Setting up a stream link

Setting up input arguments:

Procedure

1. Click the **Input Arguments** tab.

One of the following conditions applies:

- If you used the Web Services Meta Data Importer to import the web services, the generated table definition created is automatically selected and loaded. A window opens if there is conflict with an existing column.
 - If you did not use the Web Services Meta Data Importer to import the web service, see “Loading table definitions.”
2. Using the Value property, supply constants or job parameters (*#param#*).

Loading table definitions:

About this task

If you did not use the Web Services Meta Data Importer to import a web service, the Table Definitions dialog box opens.

The system highlights the table definition for the input message (*operation_IN*) of the operation that is listed on the **General** tab of the **Stage** properties page.

To use the *operation_IN* table definition:

Procedure

1. Accept the default table definition (*operation_IN*) selection by clicking **OK**.
The Select Columns dialog box opens.
2. Accept the column selections by clicking **OK**.
The namespace information is populated, and the selected input columns are created.

Setting up a reference link

About this task

To set up input arguments with a reference link:

Procedure

1. Click the **Input Arguments** tab.
The properties include Lookup, through which you indicate that a column on the reference link supplies the input value.
2. To load namespace information and input parameters for the web service operation listed as a Stage property, click **Load Arguments Information**.
The Table Definitions dialog box opens. The system highlights the table definition for the input message (*operation_IN*) of the operation that is listed on the **General** tab of the **Stage** properties page.

Note: If you did not use the Web Service Meta Data Importer, the table definition for the input message is not highlighted.

3. Accept the default table definition (*operation_IN*) selection by clicking **OK**. The Select Columns dialog box opens.
4. Accept the column selections by clicking **OK**. The namespace information is populated, and the selected input columns are created.
5. Locate the input columns whose values are supplied by the reference link.
6. In the Value cells, specify the names of the columns on the reference link that supply the values.
7. Select the corresponding **Lookup** check boxes.

Supplying an input SOAP header

About this task

Application: Output links that are either stream or reference links.

To supply an input SOAP header:

Procedure

1. Click the **Input Arguments** tab.
2. In the grid, right-click and select **Insert request value**. A row is added.
3. In the Name column, enter a meaningful description, such as Header. The description is not passed to the web service.
4. In the Value column, enter one of the following values:
 - XML chunk that consists of input header elements
 - Job parameter (*#param#*) that supplies the XML chunk
5. Select the **Header** check box.
6. Leave the Mapping column blank.

When the server job runs, Web Services Pack generates the SOAP header block. For example:

```
<soapenv:Envelope...>
  <soapenv:Header>
    <user>WSUser</user>
  </soapenv:Header>
  <soapenv:Body>
    ...
  </soapenv:Body>
</soapenv:Envelope>
```

Setting up Output Message properties

Use the Output Message page to perform one of these actions:

- Load namespace information and output parameters from the table definition that contains WSDL information. The Web Services Client stage uses this information to create an output message.
- Specify the column on the output link that receives the response from the web service.

If you want the Web Services Client stage to parse the output message and map the output parameters to InfoSphere DataStage columns, see Loading Message Information from the Web Service.

If you want the Web Services Client stage to pass the output message as is to a single column, see [Returning Output Messages Without Decoding](#).

Loading message information from the Web Service

About this task

To load message information from the web service:

Procedure

1. Click the **Output Message** tab.
2. Click **Load Message Information**.

One of the following conditions applies:

- If you used the Web Services Meta Data Importer to import the web services, the generated table definition created is automatically selected and loaded. A window opens if there is conflict with an existing column.
- If you did not use the Web Services Meta Data Importer to import the web service, see [Loading Table Definitions](#).

Loading table definitions:

About this task

If you did not use the Web Services Meta Data Importer to import the web service, the Table Definitions dialog box opens.

The system highlights the table definition for the output message (*operation_OUT*) of the operation that is listed on the **General** tab of the **Stage** properties page.

To load the *operation_OUT* table definition:

Procedure

1. Accept the default table definition (*operation_OUT*) selection by clicking **OK**.
The Select Columns dialog box opens.
2. Accept the column selections by clicking **OK**.
The namespace information is populated, and the selected output columns are created.

Returning output messages without decoding

About this task

To return output messages without decoding:

Procedure

1. Define a column in the input link of the receiving stage.
2. On the Output page of the Web Services Client stage, click the **Output Message** tab.
3. Select the **User-Defined Message** check box.
4. In the **Choose the Column Receiving the User Message** list, select the column of the linked stage that will receive the output message.

Note: If you configure the Web Services Client stage or Web Services Transformer stage to handle user-defined messages or headers, you must define the column containing the message as VarBinary, especially if the message is not UTF-8 encoded.

Setting up Output Header properties

About this task

Use the Output Header page to specify an output column that receives the output header that is sent in a web service response.

Procedure

1. Click the **Output Header** tab.
2. Select the **User-Defined Header** check box.
3. In the **Choose the Column Receiving the User Header** list, select the output column that receives the output header.

Maintaining Columns properties

About this task

Use the Columns page to:

- Inspect the definitions of output values.
- Load another table definition. This is an advanced task.

Procedure

1. Click the **Columns** tab.

The rows contain one of the following values:

- Parameters from the output message, with XPath expressions in the Description column
 - Name of the column from a linked stage that receives the output message
2. Modify the information, as needed.

Chapter 5. Creating Web Service routines

An IBM InfoSphere DataStage Server routine can call a web service operation. These web service routines work as Transform functions, which you can use with the built-in Transformer stage.

About input arguments

You can call operations that require at least one input parameter. Web service input parameters map to the input arguments of an IBM InfoSphere DataStage routine, as follows:

- Atomic values map as is.
- Complex data structures are flattened. For example, an integer and a structure that contains a string and a float map to three input arguments: integer, string, float.

Unsupported input processing

Routines do not accept the following items:

- SOAP messages that are encoded by other stages
- Arrays
- Proxy servers
- SOAP headers
- Basic HTTP authentication
- HTTPS communication

About return values

You can call a web service that returns zero or more output parameters.

Web service responses map to return values of an IBM InfoSphere DataStage routine, as follows:

- Atomic values map as is.
- Complex data structures are flattened and mapped to InfoSphere DataStage dynamic arrays.

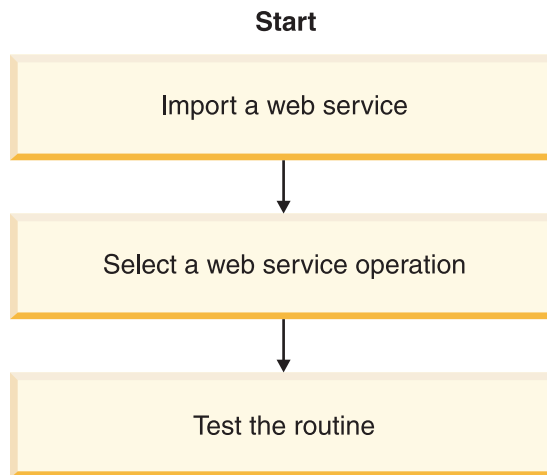
Unsupported output processing

Routines do not support the following items:

- Passing the response SOAP message without decoding to another stage. The routine decodes responses.
- Arrays

Required tasks in Web Service routines

The following diagram illustrates the tasks needed to generate a web service routine.



Importing a Web Service

The first step in using web service operations is importing the web service. For more information, see [Using the Web Service Meta Data Importer](#).

Selecting a Web Service operation

About this task

A web service routine calls a single operation. To select the operation, you use the **Web Service Browser**, which you access through the InfoSphere DataStage Designer.

Procedure

1. Open the Designer client.
2. Select **Import > Web Service Function Definitions**.
The **Web Service Meta Data Repository Browser** opens.
3. Select an operation. For more information about selecting an operation, see [Setting up General Properties](#).

The web service name is used as a routine category in the InfoSphere DataStage repository browser:

```
Routines\WebServices\service_name
```

In the right window, the operation name is appended to the web service name using a period (.) to form the name of the routine.

Results

If the operation or web service name contains characters that are not supported in a routine name, such as spaces, a dialog box opens in which you can enter a valid routine name.

Examining the Web Service routine

This section explores web service information that is presented in the Server Routine dialog box.

About the General page

The General page contains the following information:

- Routine name (operation name, appended to the web service name)
- Web service endpoint URL (Short description)
- Documentation of the operation that the WSDL provides (Long description)

About the Arguments page

The Arguments page lists the input parameters for the operation.

About the Code page

The Code page presents the UNIBASIC code for the routine. The code includes logic for mapping SOAP messages, calling the web service, and handling errors.

About the Dependencies page

The Dependencies page contains the following information:

- Name of the routine (Name)
- Type of dependency, which is always Web Service (Type)
- Web service URL (Location)

Testing the Web Service routine

About this task

You can test the routine by providing single values for each input parameter.

Procedure

1. On the Code page, click **Test**.
The Test Routine dialog box opens.
2. Enter values for each input parameter.
3. Click **Run** to run the routine.
The first line of the web service response opens in the Results column of the Test Routine dialog box.
4. To access the full response, double-click the **Results** entry.

Appendix A. Product accessibility

You can get information about the accessibility status of IBM products.

The IBM InfoSphere Information Server product modules and user interfaces are not fully accessible.

For information about the accessibility status of IBM products, see the IBM product accessibility information at http://www.ibm.com/able/product_accessibility/index.html.

Accessible documentation

Accessible documentation for InfoSphere Information Server products is provided in an information center. The information center presents the documentation in XHTML 1.0 format, which is viewable in most web browsers. Because the information center uses XHTML, you can set display preferences in your browser. This also allows you to use screen readers and other assistive technologies to access the documentation.

The documentation that is in the information center is also provided in PDF files, which are not fully accessible.

IBM and accessibility

See the IBM Human Ability and Accessibility Center for more information about the commitment that IBM has to accessibility.

Appendix B. Contacting IBM

You can contact IBM for customer support, software services, product information, and general information. You also can provide feedback to IBM about products and documentation.

The following table lists resources for customer support, software services, training, and product and solutions information.

Table 1. IBM resources

Resource	Description and location
IBM Support Portal	You can customize support information by choosing the products and the topics that interest you at www.ibm.com/support/entry/portal/Software/Information_Management/InfoSphere_Information_Server
Software services	You can find information about software, IT, and business consulting services, on the solutions site at www.ibm.com/businesssolutions/
My IBM	You can manage links to IBM Web sites and information that meet your specific technical support needs by creating an account on the My IBM site at www.ibm.com/account/
Training and certification	You can learn about technical training and education services designed for individuals, companies, and public organizations to acquire, maintain, and optimize their IT skills at http://www.ibm.com/training
IBM representatives	You can contact an IBM representative to learn about solutions at www.ibm.com/connect/ibm/us/en/

Appendix C. Accessing the product documentation

Documentation is provided in a variety of formats: in the online IBM Knowledge Center, in an optional locally installed information center, and as PDF books. You can access the online or locally installed help directly from the product client interfaces.

IBM Knowledge Center is the best place to find the most up-to-date information for InfoSphere Information Server. IBM Knowledge Center contains help for most of the product interfaces, as well as complete documentation for all the product modules in the suite. You can open IBM Knowledge Center from the installed product or from a web browser.

Accessing IBM Knowledge Center

There are various ways to access the online documentation:

- Click the **Help** link in the upper right of the client interface.
- Press the F1 key. The F1 key typically opens the topic that describes the current context of the client interface.

Note: The F1 key does not work in web clients.

- Type the address in a web browser, for example, when you are not logged in to the product.

Enter the following address to access all versions of InfoSphere Information Server documentation:

```
http://www.ibm.com/support/knowledgecenter/SSZJPZ/
```

If you want to access a particular topic, specify the version number with the product identifier, the documentation plug-in name, and the topic path in the URL. For example, the URL for the 11.3 version of this topic is as follows. (The ⇒ symbol indicates a line continuation):

```
http://www.ibm.com/support/knowledgecenter/SSZJPZ_11.3.0/⇒  
com.ibm.swg.im.iis.common.doc/common/accessingiidoc.html
```

Tip:

The knowledge center has a short URL as well:

```
http://ibm.biz/knowctr
```

To specify a short URL to a specific product page, version, or topic, use a hash character (#) between the short URL and the product identifier. For example, the short URL to all the InfoSphere Information Server documentation is the following URL:

```
http://ibm.biz/knowctr#SSZJPZ/
```

And, the short URL to the topic above to create a slightly shorter URL is the following URL (The ⇒ symbol indicates a line continuation):

```
http://ibm.biz/knowctr#SSZJPZ_11.3.0/com.ibm.swg.im.iis.common.doc/⇒  
common/accessingiidoc.html
```

Changing help links to refer to locally installed documentation

IBM Knowledge Center contains the most up-to-date version of the documentation. However, you can install a local version of the documentation as an information center and configure your help links to point to it. A local information center is useful if your enterprise does not provide access to the internet.

Use the installation instructions that come with the information center installation package to install it on the computer of your choice. After you install and start the information center, you can use the **iisAdmin** command on the services tier computer to change the documentation location that the product F1 and help links refer to. (The `⇒` symbol indicates a line continuation):

Windows

```
IS_install_path\ASBServer\bin\iisAdmin.bat -set -key ⇒  
com.ibm.iis.infocenter.url -value http://<host>:<port>/help/topic/
```

AIX® Linux

```
IS_install_path/ASBServer/bin/iisAdmin.sh -set -key ⇒  
com.ibm.iis.infocenter.url -value http://<host>:<port>/help/topic/
```

Where `<host>` is the name of the computer where the information center is installed and `<port>` is the port number for the information center. The default port number is 8888. For example, on a computer named `server1.example.com` that uses the default port, the URL value would be `http://server1.example.com:8888/help/topic/`.

Obtaining PDF and hardcopy documentation

- The PDF file books are available online and can be accessed from this support document: <https://www.ibm.com/support/docview.wss?uid=swg27008803&wv=1>.
- You can also order IBM publications in hardcopy format online or through your local IBM representative. To order publications online, go to the IBM Publications Center at <http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss>.

Appendix D. Providing feedback on the product documentation

You can provide helpful feedback regarding IBM documentation.

Your feedback helps IBM to provide quality information. You can use any of the following methods to provide comments:

- To provide a comment about a topic in IBM Knowledge Center that is hosted on the IBM website, sign in and add a comment by clicking **Add Comment** button at the bottom of the topic. Comments submitted this way are viewable by the public.
- To send a comment about the topic in IBM Knowledge Center to IBM that is not viewable by anyone else, sign in and click the **Feedback** link at the bottom of IBM Knowledge Center.
- Send your comments by using the online readers' comment form at www.ibm.com/software/awdtools/rcf/.
- Send your comments by e-mail to comments@us.ibm.com. Include the name of the product, the version number of the product, and the name and part number of the information (if applicable). If you are commenting on specific text, include the location of the text (for example, a title, a table number, or a page number).

Notices and trademarks

This information was developed for products and services offered in the U.S.A. This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

Notices

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session or persistent cookies. If a product or component is not listed, that product or component does not use cookies.

Table 2. Use of cookies by InfoSphere Information Server products and components

Product module	Component or feature	Type of cookie that is used	Collect this data	Purpose of data	Disabling the cookies
Any (part of InfoSphere Information Server installation)	InfoSphere Information Server web console	<ul style="list-style-type: none"> • Session • Persistent 	User name	<ul style="list-style-type: none"> • Session management • Authentication 	Cannot be disabled
Any (part of InfoSphere Information Server installation)	InfoSphere Metadata Asset Manager	<ul style="list-style-type: none"> • Session • Persistent 	No personally identifiable information	<ul style="list-style-type: none"> • Session management • Authentication • Enhanced user usability • Single sign-on configuration 	Cannot be disabled

Table 2. Use of cookies by InfoSphere Information Server products and components (continued)

Product module	Component or feature	Type of cookie that is used	Collect this data	Purpose of data	Disabling the cookies
InfoSphere DataStage	Big Data File stage	<ul style="list-style-type: none"> • Session • Persistent 	<ul style="list-style-type: none"> • User name • Digital signature • Session ID 	<ul style="list-style-type: none"> • Session management • Authentication • Single sign-on configuration 	Cannot be disabled
InfoSphere DataStage	XML stage	Session	Internal identifiers	<ul style="list-style-type: none"> • Session management • Authentication 	Cannot be disabled
InfoSphere DataStage	IBM InfoSphere DataStage and QualityStage Operations Console	Session	No personally identifiable information	<ul style="list-style-type: none"> • Session management • Authentication 	Cannot be disabled
InfoSphere Data Click	InfoSphere Information Server web console	<ul style="list-style-type: none"> • Session • Persistent 	User name	<ul style="list-style-type: none"> • Session management • Authentication 	Cannot be disabled
InfoSphere Data Quality Console		Session	No personally identifiable information	<ul style="list-style-type: none"> • Session management • Authentication • Single sign-on configuration 	Cannot be disabled
InfoSphere QualityStage Standardization Rules Designer	InfoSphere Information Server web console	<ul style="list-style-type: none"> • Session • Persistent 	User name	<ul style="list-style-type: none"> • Session management • Authentication 	Cannot be disabled
InfoSphere Information Governance Catalog		<ul style="list-style-type: none"> • Session • Persistent 	<ul style="list-style-type: none"> • User name • Internal identifiers • State of the tree 	<ul style="list-style-type: none"> • Session management • Authentication • Single sign-on configuration 	Cannot be disabled
InfoSphere Information Analyzer	Data Rules stage in the InfoSphere DataStage and QualityStage Designer client	Session	Session ID	Session management	Cannot be disabled

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)[®] are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at www.ibm.com/legal/copytrade.shtml.

The following terms are trademarks or registered trademarks of other companies:

Adobe is a registered trademark of Adobe Systems Incorporated in the United States, and/or other countries.

Intel and Itanium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java[™] and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

The United States Postal Service owns the following trademarks: CASS, CASS Certified, DPV, LACS^{Link}, ZIP, ZIP + 4, ZIP Code, Post Office, Postal Service, USPS and United States Postal Service. IBM Corporation is a non-exclusive DPV and LACS^{Link} licensee of the United States Postal Service.

Other company, product or service names may be trademarks or service marks of others.

Index

A

- active stage, see Web Services Transformer Stage 5
- Advanced button 22, 36
- Arguments page
 - Server Routine 47
- ASMX files
 - accessing WSDLs through 9
 - limitations 8
- ASP.NET Web Services files, see ASMX files 7

B

- binding (WSDL element) 4
- buttons
 - Advanced 22, 36
 - Close 10
 - Details 10
 - Load Arguments Information 41
 - Load Message Information 24, 26, 38, 43
 - Open 9
 - Run 47
 - Select Web Service Operation 21, 36
 - Test 47

C

- Close button 10
- Code page
 - Server Routine 47
- Columns page
 - Input Link properties
 - Web Services Client Stage 40
 - Web Services Transformer Stage 25
 - Output Link properties
 - Web Services Client Stage 44
 - Web Services Transformer Stage 27
- configuring
 - Web Services Client Stage 30
- creating table definitions 10
- customer support
 - contacting 51

D

- Dependencies page
 - Server Routine 47
- Details button 10
- dialog boxes
 - Import Progress 10
 - Open Local or UNC File 9
 - Select Columns 25, 27, 39, 41, 42, 43
 - Server Routine 46
 - Table Definitions 24, 39, 41, 43
 - Test Routine 47

- dialog boxes (*continued*)
 - Web service information 22, 36
 - Web Services Client Stage 36
 - Web Services Transformer Stage 21, 22
- DISCO files
 - accessing WSDLs through 9
 - limitations 8
- Discovery of Web Services files, see DISCO files 7
- document-style 7

E

- encodings 7
- encodingStyle attribute 3
- encryption, SSL 23, 37
- errors, import
 - accessing tracing information 11
 - accessing XML Meta Data Importer 11
- copying 11
- viewing 11

G

- General page
 - Input Link properties
 - Web Services Client Stage 38
 - Web Services Transformer Stage 24
 - Output Link properties
 - Web Services Client Stage 40
 - Web Services Transformer Stage 26
 - Server Routine 47
 - Stage properties
 - Web Services Client Stage 36
 - Web Services Transformer Stage 21
- green pages (UDDI category) 4

H

- header
 - SOAP 18, 33
- home page, Web Services Meta Data Importer 13
- HTTP security 19, 23, 33, 37
- HTTPS security 19, 33

I

- Import Progress dialog box 10
- importing
 - web service operation 11
 - web services 15, 31, 46
- input arguments
 - in web service routines 45

- Input Arguments page
 - Output Link properties
 - Web Services Client Stage 40
- Input Header page
 - Input Link properties
 - Web Services Client Stage 39
 - Web Services Transformer Stage 25
- Input Link properties
 - Web Services Client Stage 38, 40
 - Web Services Transformer Stage 23, 26
- Input Message page
 - Input Link properties
 - Web Services Client Stage 38
 - Web Services Transformer Stage 24
- input messages
 - identifying 10
 - importing 7
- input values
 - supplied by reference links 35

L

- legal notices 57
- Load Arguments Information button 41
- Load Message Information button 24, 26, 38, 43
- loading message information 43
 - without decoding 27
- logging SOAP faults 19

M

- mapper table definitions 12
- mapping metadata 10
- message (WSDL element) 4
- message information
 - loading without decoding 27
- metadata
 - mapping 10
 - table definitions 12
 - viewing 22, 36
- modifying web service request 20

O

- Open button 9
- Open Local or UNC File dialog box 9
- operation
 - see web service operation 1
- Options page
 - Stage properties
 - Web Services Client Stage 37
 - Web Services Transformer Stage 22
- Output Header page
 - Output Link properties
 - Web Services Client Stage 44

- Output Header page *(continued)*
 - Output Link properties *(continued)*
 - Web Services Transformer Stage 27
- Output Link properties
 - Web Services Client Stage 40, 44
 - Web Services Transformer Stage 26, 28
- Output Message page
 - Output Link properties
 - Web Services Client Stage 42
 - Web Services Transformer Stage 26
- output messages, identifying 10

P

- parsing an output message 42
- passing an output message as is 43
- passive stage, see Web Services Client stage 5
- passthrough data from input to output 19
- port type
 - as WSDL element 4
 - viewing 10
- private registry 4
- processing single operations 11
- product accessibility
 - accessibility 49
- product documentation
 - accessing 53
- properties
 - Web Services Client Stage 35, 44
 - Web Services Transformer Stage 21, 28
- Proxy page
 - Stage properties
 - Web Services Transformer Stage 23
 - Stage Properties
 - Web Services Client Stage 37
- proxy server 19, 34
- public registry 4

R

- reference links
 - supplying input values 35
- registry, web service 4
- Reject link 20
- return values
 - atomic values 45
 - complex data structures 45
 - web service routines 45
- returning message information
 - without decoding 43
- routines, see web service routines 45
- RPC-style 7
- Run button 47

S

- security
 - HTTP 19, 23, 33, 37
 - HTTPS 19, 33

- Security page
 - Stage properties
 - Web Services Client Stage 37
 - Web Services Transformer Stage 23
- Select Columns dialog box 25, 27, 39, 41, 42, 43
- Select Web Service Operation button 21, 36
- Server Routine
 - Arguments page 47
 - Code page 47
 - Dependencies page 47
 - General page 47
- Server Routine dialog box 46
- service (WSDL element) 4
- Simple Object Access Control, see SOAP 2
- SOAP
 - body 3
 - definition 2
 - envelope 3
 - faults 19, 20, 34, 37
 - header 3, 18, 33
 - message
 - example 3
 - from a previous stage 25, 39
 - importing 7
 - structure 3
 - over HTTP 3
- software services
 - contacting 51
- SSL encryption 23, 37
- stack 2
- Stage properties
 - Web Services Client Stage 35, 38
 - Web Services Transformer Stage 21
- stream link 41
- support
 - customer 51

T

- table definitions
 - creating 7, 10
 - mapper 12
 - metadata 12
- Table Definitions dialog box 24, 39, 41, 43
- technology stack 2
- Test button 47
- Test Routine dialog box 47
- time-out factor 19, 20, 34, 37
- trademarks
 - list of 57
- transport protocols, bindings 3

U

- UBR
 - data 4
 - definition 4
- UDDI
 - category
 - green pages 4
 - white pages 4

- UDDI *(continued)*
 - category *(continued)*
 - yellow pages 4
 - description 4
 - directory 4
- UDDI Business Registry
 - see UBR 4
- Universal Description, Discovery, and Integration, see UDDI 4

V

- viewing metadata 22, 36

W

- W3C
 - SOAP specifications 3
- web service
 - as data source 29, 32, 35, 40
 - as data target 29, 32, 35, 38
 - definition 1
 - importing 15, 31, 46
 - protocols and standards 2
 - SOAP header 18, 33
 - technology stack 2
- Web service information dialog box 22, 36
- Web Service Meta Data Importer
 - starting 22
- Web Service Meta Data Repository
 - Browser 46
- web service operation
 - definition 1
 - importing 10, 11
 - processing singly 11
 - selecting 16, 31
 - selecting for web service routines 46
- web service request 20
 - configuring 16, 31
 - entering manually 20, 34
 - handling arrays of input values 16
 - modifying 34
 - time-out factor 20, 34
- web service response, configuring 17, 32
- web service routine
 - creating 46
 - description 45
 - selecting an operation 46
 - testing 47
- Web Services Client Stage
 - adding to server job 35
 - configuring 30
 - Input Link properties 38, 40
 - Columns page 40
 - General page 38
 - Input Header page 39
 - Input Message page 38
 - Output Link properties 40, 44
 - Columns page 44
 - General page 40
 - Input Arguments page 40
 - Output Header page 44
 - Output Message page 42
 - Stage properties 35, 38
 - General page 36

- Web Services Client Stage (*continued*)
 - Stage properties (*continued*)
 - Options page 37
 - Proxy page 37
 - Security page 37
- Web Services Client Stage dialog box 36
- Web Services Description Language, see WSDL 4
- Web Services Pack, description 5
- Web Services Transformer Stage
 - dialog box 21, 22
 - Input Link properties 23, 26
 - Columns page 25
 - General page 24
 - Input Header page 25
 - Input Message page 24
 - Output Link properties 26, 28
 - Columns page 27
 - General page 26
 - Output Header page 27
 - Output Message page 26
 - Stage properties 21
 - General page 21
 - Options page 22
 - Proxy page 23
 - Security page 23
- white pages (UDDI category) 4
- Worldwide Web Consortium, see W3C 3
- wrapper element, see SOAP envelope 3
- WSDL
 - description 4
 - documents
 - contents 4
 - importing data from 7
 - importing from 9
 - elements 4
 - limitations 8

X

- XML Meta Data Importer 11

Y

- yellow pages (UDDI category) 4



Printed in USA

SC19-4289-00

