SAP Fieldglass

PUBLIC

SAP Fieldglass

2021-07-13

SAP Fieldglass Web Service Technical Specification

Configuration Guide

THE BEST RUN SAP

Content

1 Introduction. 3

2 About This Guide. 5

3 Request Types and Data Transfers 6

4 Request File Formats and Structures. 8

4.1 dataTransfer. 9

4.2 dataTransferStatus. 11

4.3 getDataByTransactionId. 12

5 Activation and Setup. 13

5.1 Download the WSDL Document. 14

5.2 Download the WSDL Schema File. 15

Schema Definition File Descriptions. 15

5.3 Examples: Data Transfer Requests and Responses. 19

Upload Request. 20

cXML Invoice Download Request with Data Marker. 20

Asynchronous cXML Invoice Download Response. 21

cXML Invoice Download Request with System ID as Connector Argument. 22

cXML Invoice Download Request with System ID as Separate Element. 23

getDataByTransactionID - cXML Invoice Request File. 24

dataTransferStatusType with transactionIdentifier. 25

Synchronous cXML Invoice Response. 25

6 Request Status Messages and Return Codes. 29

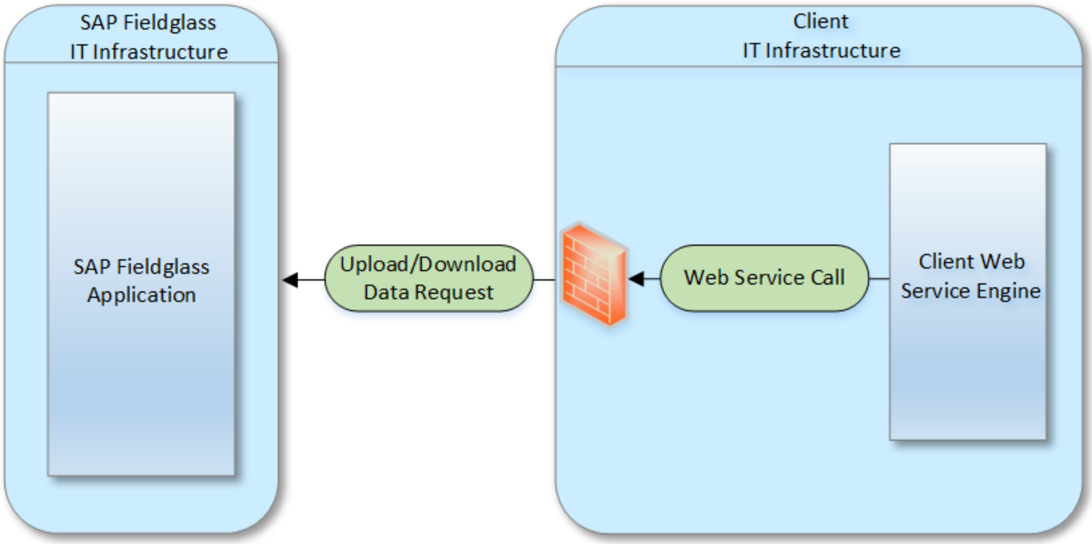
1 Introduction

The SAP Fieldglass Web Service Technical Specification is a proprietary, Web Service Description Language (WSDL) document that allows clients to download and upload data seamlessly between the application and a client system.

A web service is a method of communication between 2 systems via the Internet. The SAP Fieldglass web service is one of 2 data transmission methods supported for connector downloads and connector uploads (the other method is SFTP, a subscription-based service). The benefit of using the web service is that it uses Secure Sockets Layer (SSL) to establish an encrypted link between the web server and a web browser, eliminating the need for an SFTP server to schedule a secure job.

When using the web service, SAP Fieldglass provides the client with an XML-based WSDL document, which essentially acts as a contract between the client system and the SAP Fieldglass application. The WSDL document describes to the client system what operations are available and how to use them.

The SAP Fieldglass application uses the web service to exchange master data and transactional data between the application and a client system using download and upload connectors, which bypass the SFTP server. The web service call runs on a client system such as a homegrown or third-party application, behind the client's firewall, as illustrated next.



The illustration shows the following workflow:

- The client/supplier system "calls" the SAP Fieldglass Web Service to trigger the respective connector.
- For an upload request using a BULK or API connector version set to Asynchronous, the call sends formatted data to SAP Fieldglass. The client system receives an HTTP response from SAP Fieldglass, which indicates the success or failure of the upload. The response is sent as soon as it's confirmed that SAP Fieldglass has successfully received the file, but the response doesn't indicate the success or failure of processing individual data records. (This isn't applicable to other connector versions.)
- For a download request, SAP Fieldglass sends data back to the client/supplier system. The data can be in XML or non-XML format. The client then extracts the data from the XML and loads it into the client system.

- If the response of an upload request or download request indicates a failure, then the return error code indicates the source of the problem. To view the errors, users with the appropriate permissions can sign in to the buyer company within SAP Fieldglass and view the file in the Integration Audit Trail, which contains a record for each individual failed record.

Related Information

About This Guide [page 5]
Request Types and Data Transfers [page 6]
Request File Formats and Structures [page 8]
Activation and Setup [page 13]
Request Status Messages and Return Codes [page 29]

2 About This Guide

This guide provides detailed configuration information for the SAP Fieldglass Web Service. The topics cover process and workflow, technical specifications, and instructions on how to activate, set up, and use the WSDL document to download and upload data. In addition, detailed code samples illustrate the request file formats and structures, and data transfer requests and responses.

Related Information

Request Types and Data Transfers [page 6]
Request File Formats and Structures [page 8]
Request Status Messages and Return Codes [page 29]
Activation and Setup [page 13]
Request Status Messages and Return Codes [page 29]

3 Request Types and Data Transfers

The SAP Fieldglass Web Service Technical Specification supports both upload and download request types. For download requests, it supports synchronous and asynchronous data transfers.

Upload Requests

Upload requests are used to upload data from a client system to the SAP Fieldglass VMS system. Upload requests support multiple data formats, including cXML, text, and XML.

When using a BULK or API version of a connector set to Asynchronous, the client system receives an HTTP response from SAP Fieldglass once it's confirmed that SAP Fieldglass has successfully received the file. The response specifies the success or failure of the upload, but it doesn't specify the success or failure of processing individual data records.

Download Requests

Download requests are used to download data from the SAP Fieldglass VMS system to a client system. Consult the connector documentation for information on the various data formats supported.

> i Note
>
> The Web Service Technical Specification incorporates an optimized structure that supports the data marker functionality used in SAP S4/HANA integration. The data markers allow you to download small batches of cXML invoices or other data by specifying either a date, an invoice count, or a transaction ID.

Synchronous and Asynchronous Data Transfers – Download Requests Only

The SOAP binding suite of the Web Service Technical Specification has 3 request formats:

- dataTransfer
- dataTransferStatus
- getDataByTransactionId

For more information on the request types, see Request File Formats and Structures [page 8].

Synchronous File Download

Synchronous file download is used to receive download data immediately in the SOAP response. It's suitable when the download request isn't complex. Connector downloads processed through the SAP Fieldglass user interface are always synchronous.

Asynchronous Data Transfers

Asynchronous file download is used to receive download data at a later time. It's suitable when the download request is complex and takes longer to complete.

Asynchronous file download uses 2 or 3 operations:

1. Download request is submitted using the `dataTransfer` operation. You can generate a unique client reference ID and associate it with the request (see dataTransfer [page 9] operation elements). The request response includes a unique SAP Fieldglass transaction ID. Download data records aren't sent in the response.
2. The status of the download request can be verified using the `dataTransferStatus` operation, which accepts both client reference ID and SAP Fieldglass transaction ID. If you associate a client reference ID with the download request, you can specify a client reference ID or SAP Fieldglass transaction ID. If you don't associate a client reference ID with the download request, you need to specify an SAP Fieldglass transaction ID. The status response indicates whether the download process completed successfully, is in progress, or failed.
3. Upon successful processing of the download request, you can invoke the `getDataByTransactionId` operation to receive the download data. This operation accepts both client reference ID and SAP Fieldglass transaction ID. If you associate a client reference ID with the download request, you can specify a client reference ID or SAP Fieldglass transaction ID. If you don't associate a client reference ID with the download request, you need to specify an SAP Fieldglass transaction ID.

> **i Note**
>
> SAP Fieldglass Professional Services can provide guidance on when to choose synchronous and asynchronous download.

Related Information

Request File Formats and Structures [page 8]
Activation and Setup [page 13]
Request Status Messages and Return Codes [page 29]

4 Request File Formats and Structures

Describes the file format elements supported by the SAP Fieldglass Web Service Technical Specification.

The SAP Fieldglass Web Service Technical Specification supports the following file format elements:

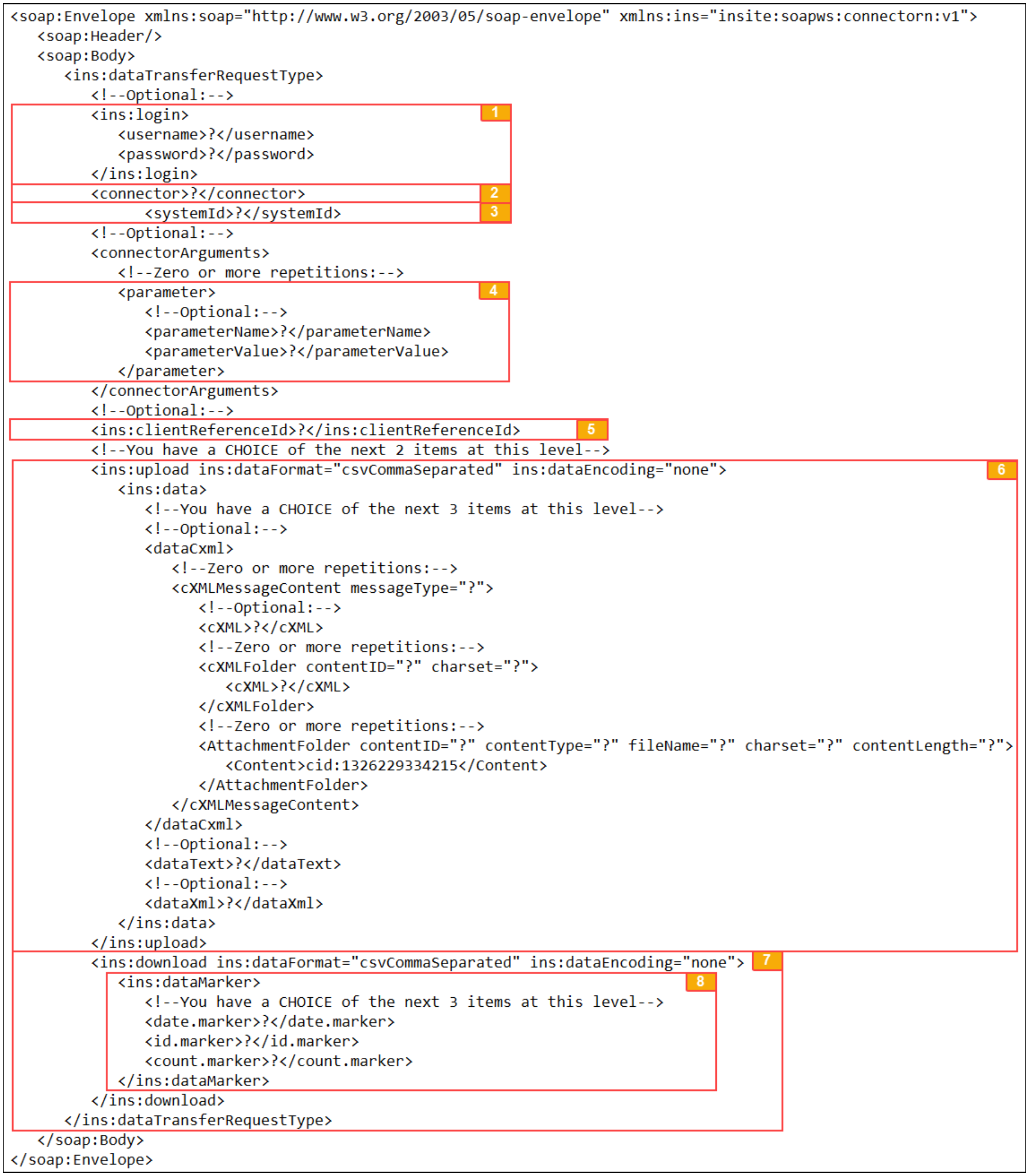
- dataTransfer
- dataTransferStatus
- getDataByTransactionId

Related Information

dataTransfer [page 9]
dataTransferStatus [page 11]
getDataByTransactionId [page 12]
Activation and Setup [page 13]
Request Status Messages and Return Codes [page 29]

4.1 dataTransfer

When using the `dataTransfer` request type, you can refer to this sample code and element structure.



```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:ins="insite:soapws:connectorn:v1">
   <soap:Header/>
   <soap:Body>
      <ins:dataTransferRequestType>
         <!--Optional:-->
         <ins:login>
            <username>?</username>
            <password>?</password>
         </ins:login>
         <connector>?</connector>
               <systemId>?</systemId>
         <!--Optional:-->
         <connectorArguments>
            <!--Zero or more repetitions:-->
            <parameter>
               <!--Optional:-->
               <parameterName>?</parameterName>
               <parameterValue>?</parameterValue>
            </parameter>
         </connectorArguments>
         <!--Optional:-->
         <ins:clientReferenceId>?</ins:clientReferenceId>
         <!--You have a CHOICE of the next 2 items at this level-->
         <ins:upload ins:dataFormat="csvCommaSeparated" ins:dataEncoding="none">
            <ins:data>
               <!--You have a CHOICE of the next 3 items at this level-->
               <!--Optional:-->
               <dataCxml>
                  <!--Zero or more repetitions:-->
                  <cXMLMessageContent messageType="?">
                     <!--Optional:-->
                     <cXML>?</cXML>
                     <!--Zero or more repetitions:-->
                     <cXMLFolder contentID="?" charset="?">
                        <cXML>?</cXML>
                     </cXMLFolder>
                     <!--Zero or more repetitions:-->
                     <AttachmentFolder contentID="?" contentType="?" fileName="?" charset="?" contentLength="?">
                        <Content>cid:1326229334215</Content>
                     </AttachmentFolder>
                  </cXMLMessageContent>
               </dataCxml>
               <!--Optional:-->
               <dataText>?</dataText>
               <!--Optional:-->
               <dataXml>?</dataXml>
            </ins:data>
         </ins:upload>
         <ins:download ins:dataFormat="csvCommaSeparated" ins:dataEncoding="none">
            <ins:dataMarker>
               <!--You have a CHOICE of the next 3 items at this level-->
               <date.marker>?</date.marker>
               <id.marker>?</id.marker>
               <count.marker>?</count.marker>
            </ins:dataMarker>
         </ins:download>
      </ins:dataTransferRequestType>
   </soap:Body>
</soap:Envelope>
```

Element Structure

1. `login` authentication – username is buyer/supplier ID and password is license key.

2. `connector` section is the name of the SAP Fieldglass connector configured for data upload or download.
3. `systemId` section is supported for clients who have multiple instances in SAP Fieldglass and multiple instances in the client system, but only one SAP Fieldglass account. Each client system may have '*n*' cost centers, where each cost center in SAP Fieldglass is identified and related to its respective client system based on the system ID. This allows clients to download a subset of invoices by filtering on the system ID. This is currently supported for cXML Invoice Download only.
    - The system ID must already be populated on all admin objects, such as sites, cost centers, business units, and so forth. If the system ID does not exist on the admin objects, then it must be populated via data upload.
    - The system ID can be defined as a connector argument or as a separate element in the `systemId` field `(<systemId>?</systemId>).`
    - If sending the system ID as a connector argument, then `systemId` field can be removed.
    - If sending the system ID as separate element, then connector argument section can be removed.
    - If `systemId` element is present, even if it is empty, and any parameter is given, then the parameter will be ignored.
    - If `systemId` element is blank, it is assumed the client wants to download all invoices and the system will return all invoices.
4. `parameters` in `connectorArguments` section allow passing parameter values for data upload or download. Consult the respective connector document for connector names and parameters.
5. `clientReferenceId` is a unique ID provided by client to retrieve the transaction response.
6. `upload` section defines the request format for upload request, such as data format, encoding, and file formats, including cXML, XML, CDATA, and textdata.
7. `download` section defines the request format of a download request, such as data format, data encoding, and data marker.
8. `dataMarker` section defines the markers that limit the number of records to download. Currently only supported for cXML Invoice Connector. Support for other connectors may be available in the future.

Related Information

Examples: Data Transfer Requests and Responses [page 19]

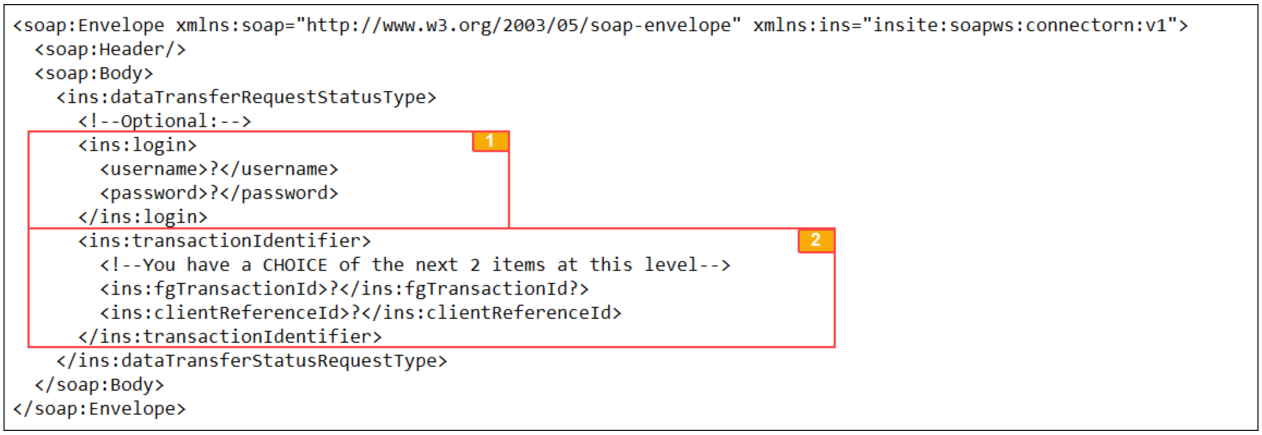
Request Status Messages and Return Codes [page 29]

cXML Invoice Download Request with System ID as Connector Argument [page 22]

cXML Invoice Download Request with System ID as Separate Element [page 23]

4.2 dataTransferStatus

When using the `dataTransferStatus` request type, you can refer to this sample code and element structure.



```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:ins="insite:soapws:connectorn:v1">
  <soap:Header/>
  <soap:Body>
    <ins:dataTransferRequestStatusType>
      <!--Optional:-->
      <ins:login>
        <username>?</username>
        <password>?</password>
      </ins:login>
      <ins:transactionIdentifier>
        <!--You have a CHOICE of the next 2 items at this level-->
        <ins:fgTransactionId>?</ins:fgTransactionId?>
        <ins:clientReferenceId>?</ins:clientReferenceId>
      </ins:transactionIdentifier>
    </ins:dataTransferStatusRequestType>
  </soap:Body>
</soap:Envelope>
```

Element Structure

1. `login` authentication – username is buyer/supplier ID and password is license key.
2. `transactionIdentifier` – specify an SAP Fieldglass transaction ID and/or client reference ID.

Related Information

Examples: Data Transfer Requests and Responses [page 19]

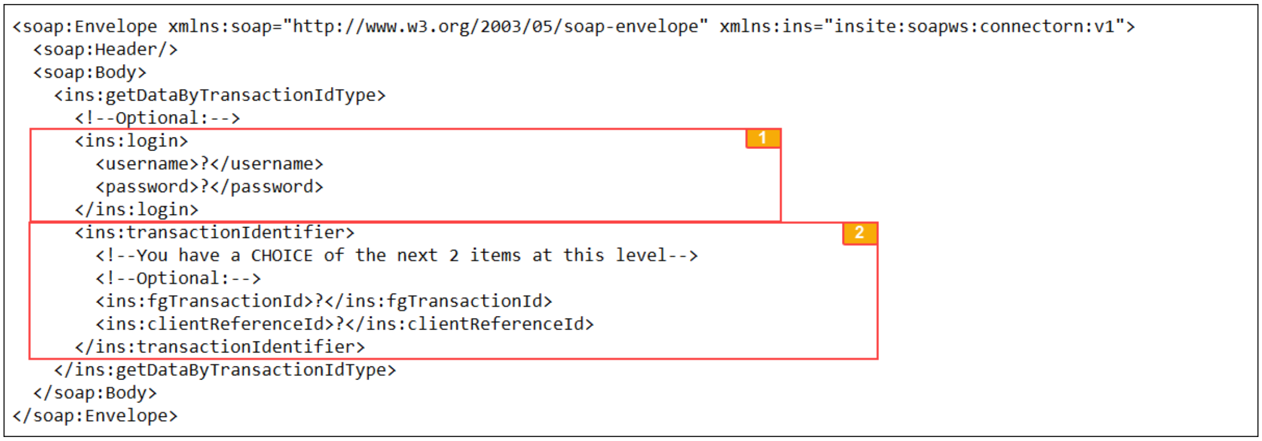
dataTransferStatusType with transactionIdentifier [page 25]

Request Status Messages and Return Codes [page 29]

4.3 getDataByTransactionId

When using the `getDataByTransactionId` request type, you can refer to this sample code and element structure.

.



```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:ins="insite:soapws:connectorn:v1">
  <soap:Header/>
  <soap:Body>
    <ins:getDataByTransactionIdType>
      <!--Optional:-->
      <ins:login>
        <username>?</username>
        <password>?</password>
      </ins:login>
      <ins:transactionIdentifier>
        <!--You have a CHOICE of the next 2 items at this level-->
        <!--Optional:-->
        <ins:fgTransactionId>?</ins:fgTransactionId>
        <ins:clientReferenceId>?</ins:clientReferenceId>
      </ins:transactionIdentifier>
    </ins:getDataByTransactionIdType>
  </soap:Body>
</soap:Envelope>
```

Element Structure

1. `login` authentication – username is buyer/supplier ID and password is license key.
2. `transactionIdentifier` – specify an SAP Fieldglass transaction ID and/or client reference ID.

Related Information

Examples: Data Transfer Requests and Responses [page 19]

getDataByTransactionID - cXML Invoice Request File [page 24]

Request Status Messages and Return Codes [page 29]

5 Activation and Setup

Describes the requirements for activating and configuring web service calls in SAP Fieldglass.

A WSDL document is an XML formatted document that describes network services as a set of endpoints that operate on messages and contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint.

The setup steps apply to invoking any connector within the library: Bulk or API, standard, or custom. Some download connectors, such as the Invoice Buyer Download, require parameters in addition to calling the connector, based on the name in the web service call. Depending on the code, custom connectors can also require additional parameters.

To activate and set up web service calls between the SAP Fieldglass VMS system and a client system, you need the following:

- WSDL documents, as described in Activation and Setup [page 13]. The client's developer needs to configure and generate the artifacts for web service calls. For information on the WSDL elements, see Schema Definition File Descriptions [page 15].
- Username and license key that you created in SAP Fieldglass. Note that the license key is the password used on the web service request.
- Endpoint URLs to post the XML in SAP Fieldglass:
  - Production endpoint: https://www.fieldglass.net/ws2/services/connectorn
  - Test endpoint: https://<env_code>.fgvms.com/ws2/services/connectorn

  > **i Note**
  >
  > The URLs are case-sensitive.

Related Information

Download the WSDL Document [page 14]
Download the WSDL Schema File [page 15]
Schema Definition File Descriptions [page 15]

5.1 Download the WSDL Document

Download the predefined WSDL document from the URL based on whether you're working in a production environment or a test environment.

The Web Service Technical Specification WSDL document contains definitions and references to the schema file. The web service calls the WSDL as needed during the upload/download process. The WSDL requires no configuration by you.

Download the latest WSDL document from SAP Fieldglass using one of the following links, based on your processing requirements:

- Production: https://www.fieldglass.net/ws2/services/connectorn?wsdl
- Test: https://<env_code>.fgvms.com/ws2/services/connectorn?wsdl (for example: https://int42.fgvms.com/ws2/services/connectorn?wsdl)

> **Note**
>
> URLs are case-sensitive. In addition, to view the actual WSDL definition, open the link in a web browser, right-click in the XML and select to view the page source, and then save and reopen the file. Otherwise, if you save the open file in the browser, you get the XML format only, not the actual WSDL definition.

The following code block shows the WSDL document definition.

Sample Code

wsdl document definition

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="insite:soapws:connectorn" xmlns:wsdl="http://
schemas.xmlsoap.org/wsdl/" xmlns:v1="insite:soapws:connectorn:v1"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:wsoap12="http://
schemas.xmlsoap.org/wsdl/soap12/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <wsdl:import namespace="insite:soapws:connectorn:v1" location="connectorn?
wsdl=v1_bindings.wsdl">
    </wsdl:import>
  <wsdl:service name="connectorn">
    <wsdl:port name="Prod" binding="v1:dataTransferSOAP12Binding">
      <wsoap12:address location="https://ws.fieldglass.net/ws2/services/
connectorn"/>
    </wsdl:port>
    <wsdl:port name="Local" binding="v1:dataTransferSOAP12Binding">
      <wsoap12:address location="http://localhost:8080/ws2/services/connectorn"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

Related Information

Download the WSDL Schema File [page 15]

Schema Definition File Descriptions [page 15]

Examples: Data Transfer Requests and Responses [page 19]

5.2 Download the WSDL Schema File

The Web Service Technical Specification schema file, `connectorn?xsd=v1_types.xsd`, contains the SAP Fieldglass data types that require configuration by the client's developer.

You can download the latest schema file from SAP Fieldglass at:

https://<*env_code*>.fgvms.com/ws2/services/connectorn?xsd=v1_types.xsd

> i Note
>
> The URL is case-sensitive.

Related Information

Schema Definition File Descriptions [page 15]
Examples: Data Transfer Requests and Responses [page 19]
Request File Formats and Structures [page 8]
Activation and Setup [page 13]

5.2.1 Schema Definition File Descriptions

Describes the elements and corresponding input parameters of the SAP Fieldglass Web Service schema definition file `connectorn?xsd=v1_types.xsd`.

Schema File Descriptions

Element Name	Description	Input Parameter
dataTransferRequestType		`login` – Requires buyer/supplier ID as username and license key as password for authentication. `connector` – Requires connector name as input value. `upload` – See `upload` element below. `download` – See `download` element below.

Element Name	Description	Input Parameter
dataTransferResponseType	Parent element for data transfer response. Data is returned in data element.	fgTransactionID – Transaction ID returned by SAP Fieldglass system. requestStatus – Includes status of the transaction with the following return parameters: statusCode, statusMessage, and retry. data – See data element below.
upload	Input elements used to specify data parameters for a data upload. Don't use for a file download.	dataFormat – Supports csvComma-Separated. See dataFormat element below. dataEncoding – None. data – See data element. Currently supports dataText. Note: textData requires data to be enclosed in <CDATA> tags to allow XML parsers to ignore special characters and process the data.
download	Input elements used to specify data parameters for a download. Format and encoding styles of downloaded data are set using dataFormat and dataEncoding attributes. Don't use for a file upload.	dataMarker – Used in download request to apply limiters on the number of records returned. Elements for data markers are: date.marker – Specific date (currently unsupported). id.marker – Transaction ID (currently unsupported). count.marker – Number of invoices/ batch size (currently supported for cXML Invoice Download). dataFormat – Currently supports cXML. dataEncoding – Optional. Valid values are: UTF-8 in Base64-wrapper UTF-8 Base64 None (default value if no value specified)

Element Name	Description	Input Parameter
getDataByTransactionIdType	Retrieves past data using `transactionIdentifier`—either client reference ID or SAP Fieldglass transaction ID.	`fgTransactionId` – Transaction ID returned by SAP Fieldglass system. `clientReferenceId` – Client reference ID must be associated with the transaction in `dataTransferRequest`.
dataTransferStatusRequestType	Retrieves the status of a data transfer request. Use either client reference ID or SAP Fieldglass transaction ID to retrieve data.	`clientReferenceId` – Must be associated with the transaction in `dataTransferRequest`. `fgTransactionId` – Returned in `dataTransferResponse`. `Status` – Can be checked for `dataTransferRequest`. Can't be checked for `dataTransferByTransactionId`.
data	Element used in data upload request and data download response.	Data encoding style specified by the `dataEncoding` element. Default data encoding is UTF-8. Not all encoding styles are supported. Consult the respective SAP Fieldglass connector document for encoding styles supported for each connector. Valid values for the data element are: text data – further categorized as: plain text csvCommaSeparated csvPipeSeparated XML data cXML data
dataEncoding	Encoding styles supported for an upload and download.	
transactionIdentifier	Transaction identifier for request.	Clients can associate their own transaction ID with a transaction using `clientReferenceId` element. SAP Fieldglass transaction ID is used, when known, and is returned in response to the initial submit of a data transfer request.

Element Name	Description	Input Parameter
dataFormat	Data format supported by the `data` element.	Valid values are: plainText csvCommaSeparated csvPipeSeparated json cXML anyXml

Comparison: SAP Fieldglass Web Service Technical Specification and Connector V3

The following table describes a comparison of the SAP Fieldglass Web Service Technical Specification and the previous WSDL version, V3.

Comparison of SAP Fieldglass Web Service Technical Specification and Connector V3

Component	SAP Fieldglass Web Service Technical Specification	Connector V3
Application	Handles upload and download request calls.	Handles upload and download request calls.
SOAP binding	The SAP Fieldglass Web Service Technical Specification SOAP binding suite has 3 request formats: dataTransfer dataTransferStatus getDataByTransactionId	The connectorV3 SOAP binding suite has 2 request formats: dataTransfer dataTransferStatus
WSDL	https://<envt>.fgvms.com/wsdl/connectorn/connectorn.wsdl Namespace revisions remain the same across multiple revisions.	https://<envt>.fgvms.com/ws2/services/ConnectorV3?wsdl
Namespace validation	Includes namespace bindings: v1_binding Versions are managed by the bindings.	Namespace binding: FgWSDLTypes_v3.0.xsd Versions are managed at service levels.

Component	SAP Fieldglass Web Service Technical Specification	Connector V3
Data Marker in dataTransferRequest for downloads	Includes the marker to limit the number of records to download. Currently only supported for cXML Invoice Connector. Support for other connectors could be available in the future.	No markers to limit the download data.
Asynchronous responses	getDataByTransactionId request can be invoked using client reference ID or transaction ID of the asynchronous requests to return responses.	Asynchronous responses can be retrieved by dataTransfer request using client reference ID.

Related Information

Examples: Data Transfer Requests and Responses [page 19]
Request Status Messages and Return Codes [page 29]

5.3 Examples: Data Transfer Requests and Responses

When coding data transfer requests and responses, you can refer to the examples for guidance.

Examples of the following requests and responses are illustrated:

- Upload Request
- cXML Invoice Download Request with Data Marker
- Response File – Asychronous cXML Invoice
- cXML Invoice Download Request with System ID as Connector Argument
- cXML Invoice Download Request with System ID as Separate Element
- getDataByTransactionID – cXML Invoice Request File
- DataTransferStatus
- Synchronous cXML Invoice Response

Related Information

Request File Formats and Structures [page 8]
Schema Definition File Descriptions [page 15]
Request Status Messages and Return Codes [page 29]

5.3.1 Upload Request

When using the SAP Fieldglass Web Service WSDL to create an Upload Request, you can refer to this sample code.

Sample Code

upload request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:ins="insite:soapws:connectorn:v1">
  <soap:Header/>
  <soap:Body>
    <ins:dataTransferRequestType>
      <!--Optional:-->
         <ins:login>
           <username>test_user</username>
           <password>test_password</password>
         </ins:login>
         <connector>QE5ABCD123_Upload Plant</connector>
         <ins:clientReferenceId>SAPZUploadSites109</ins:clientReferenceId>
         <!--You have a CHOICE of the next 2 items at this level-->
         <ins:upload ins:dataFormat="csvCommaSeparated" ins:dataEncoding="none">
           <ins:data>
             <dataText><![CDATA[Code,Name,Description,Currency,Division
Code,Country,Address 1,State/Province,ZIP/Postal Code,Address 2,Allow Candidate
Anonymity?,City,Corporate Calendar,Hours per Day,Hours per Week,Invoice Tax
Information Code,Invoice Tax Information Required?,Parent Site,Selectable Site?
1010,Plant 1 DE,Plant 1 DE,EUR,1010,DEU,Plant 1 DE,IL,65432,,,Waldorf,,,,,,,
1710,Plant 1 US,Plant 1 US,USD,1710,USA,Plant 1 US,IL,60606,,,Texas,,,,,,,
]]></dataText>
           </ins:data>
         </ins:upload>
    </ins:dataTransferRequestType>
  </soap:Body>
</soap:Envelope>
```

Related Information

Request File Formats and Structures [page 8]

Schema Definition File Descriptions [page 15]

5.3.2 cXML Invoice Download Request with Data Marker

When using the SAP Fieldglass Web Service WSDL to create a cXML Invoice Download Request with the `data.Marker` element, you can refer to this sample code.

Sample Code

cXML invoice download request with `data.Marker` element

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:ins="insite:soapws:connectorn:v1">
```

```
   <soap:Header/>
   <soap:Body>
      <ins:dataTransferRequestType>
         <!--Optional:-->
         <ins:login>
            <username>test_user</username>
            <password>test_password</password>
         </ins:login>
         <connector>cXML Formatted Invoice(s)</connector>
         <!--Optional:-->
         <ins:clientReferenceId>clienRef</ins:clientReferenceId>
         <connectorArguments>
            <!--Zero or more repetitions:-->
            <parameter>
               <!--Optional:-->
               <parameterName>systemID</parameterName>
               <parameterValue>ABCD01234567</parameterValue>
            </parameter>
         </connectorArguments>
         <ins:download ins:dataFormat="cXML" ins:dataEncoding="none">
           <ins:dataMarker>
              <count.marker>1</count.marker>
           </ins:dataMarker>
         </ins:download>
      </ins:dataTransferRequestType>
   </soap:Body>
</soap:Envelope>
```

Related Information

Request File Formats and Structures [page 8]

Schema Definition File Descriptions [page 15]

5.3.3 Asynchronous cXML Invoice Download Response

When using the SAP Fieldglass Web Service WSDL to create an Asynchronous cXML Invoice Download Response, you can refer to this sample code.

Sample Code

asynchronous cXML invoice download response

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
   <soapenv:Header/>
   <soapenv:Body>
      <ns2:dataTransferResponseType xmlns:ns2="insite:soapws:connectorn:v1">
         <ns2:fgTransactionId>z16082703320188200015981</ns2:fgTransactionId>
         <ns2:data/>
      </ns2:dataTransferResponseType>
   </soapenv:Body>
</soapenv:Envelope>
```

i Note

The response file does not require login information.

Related Information

Request File Formats and Structures [page 8]
Schema Definition File Descriptions [page 15]

5.3.4 cXML Invoice Download Request with System ID as Connector Argument

When using the SAP Fieldglass Web Service WSDL to create a cXML Invoice Download Request with `systemId` as a connector argument, you can refer to this sample code.

Sample Code

cXML invoice download request with system id as connector argument

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:ins="insite:soapws:connectorn:v1">
   <soap:Header/>
   <soap:Body>
      <ins:dataTransferRequestType>
         <!--Optional:-->
         <ins:login>
            <username>test_user</username>
            <password>test_password</password>
         </ins:login>
         <connector>cXML Formatted Invoice(s)</connector>
         <!--Optional:-->
         <connectorArguments>
            <!--Zero or more repetitions:-->
            <parameter>
               <!--Optional:-->
               <parameterName>param1</parameterName>
               <parameterValue>ABCD111011</parameterValue>
            </parameter>
         </connectorArguments>
         <!--Optional:-->
         <ins:clientReferenceId>1234567890</ins:clientReferenceId>
         <!--You have a CHOICE of the next 2 items at this level-->
         <ins:download ins:dataFormat="csvCommaSeparated"
ins:dataEncoding="none">
             <ins:dataMarker>
               <!--You have a CHOICE of the next 3 items at this level-->
               <date.marker>?</date.marker>
               <id.marker>?</id.marker>
               <count.marker>12</count.marker>
            </ins:dataMarker>
         </ins:download>
      </ins:dataTransferRequestType>
   </soap:Body>
</soap:Envelope>
```

Related Information

Request File Formats and Structures [page 8]

Schema Definition File Descriptions [page 15]

5.3.5 cXML Invoice Download Request with System ID as Separate Element

When using the SAP Fieldglass Web Service WSDL to create a cXML Invoice Download Request with `systemId` as a separate element, you can refer to this sample code.

Sample Code

cXML invoice download request with `systemId` as separate element

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:ins="insite:soapws:connectorn:v1">
   <soap:Header/>
   <soap:Body>
      <ins:dataTransferRequestType>
         <!--Optional:-->
         <ins:login>
            <username>test_user</username>
            <password>test_password</password>
         </ins:login>
         <connector>cXML Formatted Invoice(s)</connector>
               <systemId>ABCD111011</systemId >
         <!--Optional:-->
         <connectorArguments>
            <!--Zero or more repetitions:-->
            <parameter>
               <!--Optional:-->
               <parameterName>param1</parameterName>
               <parameterValue>ABCD111011</parameterValue>
            </parameter>
         </connectorArguments>
         <!--Optional:-->
         <ins:clientReferenceId>1234567890</ins:clientReferenceId>
         <!--You have a CHOICE of the next 2 items at this level-->
         <ins:download ins:dataFormat="csvCommaSeparated"
ins:dataEncoding="none">
            <ins:dataMarker>
               <!--You have a CHOICE of the next 3 items at this level-->
               <date.marker>?</date.marker>
               <id.marker>?</id.marker>
               <count.marker>12</count.marker>
            </ins:dataMarker>
         </ins:download>
      </ins:dataTransferRequestType>
   </soap:Body>
</soap:Envelope>
```

i Note

If system ID is defined as a separate element, then connector argument is ignored.

Related Information

Request File Formats and Structures [page 8]

Schema Definition File Descriptions [page 15]

5.3.6 getDataByTransactionID - cXML Invoice Request File

When using the SAP Fieldglass Web Service WSDL to create a cXML Invoice Request file with the `getDataByTransactionId` element, you can refer to this sample code.

Sample Code

cXML invoice request file with `getDataByTransactionId` element

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:ins="insite:soapws:connectorn:v1">
  <soap:Header/>
  <soap:Body>
    <ins:getDataByTransactionIdType>
      <!--Optional:-->
      <ins:login>
        <username>test_user</username>
        <password>test_password</password>
      </ins:login>
      <ins:transactionIdentifier>
        <!--You have a CHOICE of the next 2 items at this level-->
        <!--Optional:-->
        <ins:fgTransactionId>z16081016320026500772981</ins:fgTransactionId>
        <ins:clientRef></ins:clientReferenceId>
      </ins:transactionIdentifier>
    </ins:getDataByTransactionIdType>
  </soap:Body>
</soap:Envelope>
```

Related Information

getDataByTransactionId [page 12]

Schema Definition File Descriptions [page 15]

5.3.7 dataTransferStatusType with transactionIdentifier

When using the SAP Fieldglass Web Service WSDL to create a dataTransferStatusType request with the `transactionIdentifier` element, you can refer to this sample code.

Sample Code

dataTransferStatusType request with transactionIdentifier element

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:ins="insite:soapws:connectorn:v1">
  <soap:Header/>
  <soap:Body>
    <ins:dataTransferRequestType>
      <!--Optional:-->
      <ins:login>
        <username>test_user</username>
        <password>test_password</password>
      </ins:login>
      <ins:transactionIdentifier>
        <!--You have a CHOICE of the next 2 items at this level-->
        <!--Optional:-->
        <ins:fgTransactionId>ABCD_cJQR8VfYeafhTQBXGEZXEW1EQrp</
ins:fgTransactionId>
        <!--Optional:-->
        <ins:clientReferenceId>USERUPLOAD</ins:clientReferenceId>
      </ins:transactionIdentifier>
    </ins:dataTransferRequestType>
  </soap:Body>
</soap:Envelope>
```

Related Information

dataTransferStatus [page 11]

Schema Definition File Descriptions [page 15]

5.3.8 Synchronous cXML Invoice Response

When using the SAP Fieldglass Web Service WSDL to create a Synchronous cXML Invoice Response file, you can refer to this sample code.

Sample Code

synchronous cXML invoice response

```
<cXML xml:lang="en" payloadID="z16082307512534225657981@fieldglass.com"
timestamp="2016-08-23T07:51:25+00:00">
    <Header>
        <From>
            <Credential domain="Fieldglass">
                <Identity>FGLS</Identity>
            </Credential>
            <Credential domain="PrivateID">
```

```
                <Identity>769</Identity>
            </Credential>
            <Credential domain="VendorID">
                <Identity>769</Identity>
            </Credential>
        </From>
        <To>
            <Credential domain="SystemID">
                <Identity>QE6CLNT910</Identity>
            </Credential>
        </To>
        <Sender>
            <Credential domain="Fieldglass">
                <Identity>FGLS</Identity>
            </Credential>
            <UserAgent>Fieldglass</UserAgent>
        </Sender>
    </Header>
    <Request deploymentMode="production">
        <InvoiceDetailRequest>
            <InvoiceDetailRequestHeader invoiceDate="2016-08-23T07:51:25+00:00"
invoiceID="SAPZCD00000007" operation="new" purpose="lineLevelCreditMemo">
                <InvoiceDetailHeaderIndicator/>
                <InvoiceDetailLineIndicator isAccountingInLine="yes"
isTaxInLine="yes" isDiscountInLine="yes" isSpecialHandlingInLine="yes"/>
                <InvoicePartner>
                    <Contact role="soldTo" addressID="P002">
                        <Name xml:lang="en">Tools US</Name>
                        <PostalAddress>
                            <Street>54321</Street>
                            <Street></Street>
                            <City>Walldorf</City>
                            <State>IL</State>
                            <PostalCode>69190</PostalCode>
                            <Country isoCountryCode="US">United States</Country>
                        </PostalAddress>
                    </Contact>
                </InvoicePartner>
                <InvoicePartner>
                    <Contact role="billTo" addressID="P002">
                        <Name xml:lang="en">Tools US</Name>
                        <PostalAddress>
                            <Street>54321</Street>
                            <Street></Street>
                            <City>Walldorf</City>
                            <State>IL</State>
                            <PostalCode>69190</PostalCode>
                            <Country isoCountryCode="US">United States</Country>
                        </PostalAddress>
                    </Contact>
                </InvoicePartner>
                <InvoicePartner>
                    <Contact role="remitTo" addressID="7941">
                        <Name xml:lang="en">769</Name>
                        <PostalAddress>
                            <Street>123 Normantown Street</Street>
                            <Street></Street>
                            <City>Paris</City>
                            <State>TX</State>
                            <PostalCode>0</PostalCode>
                            <Country isoCountryCode="US">United States</Country>
                        </PostalAddress>
                    </Contact>
                    <IdReference domain="federalTaxID" identifier=""/>
                </InvoicePartner>
                <DocumentReference
payloadID="z16082307455049867863981@fieldglass.com"/>
```

```
                <Period startDate="2016-08-15T00:00:00+00:00"
endDate="2016-08-21T00:00:00+00:00"/>
<Extrinsic name="AribaNetwork.LegacyInvoices">Yes</Extrinsic>
            </InvoiceDetailRequestHeader>
            <InvoiceDetailOrder>
                <InvoiceDetailOrderInfo>
                    <OrderIDInfo orderID="2345678"></OrderIDInfo>
                </InvoiceDetailOrderInfo>
                <InvoiceDetailItem invoiceLineNumber="1" quantity="-1250">
                    <UnitOfMeasure>EA</UnitOfMeasure>
                    <UnitPrice>
                        <Money currency="USD">1.00</Money>
                    </UnitPrice>
                    <InvoiceDetailItemReference lineNumber="1">
                        <ItemID>
                            <SupplierPartID>SAPZWO00000004</SupplierPartID>
                        </ItemID>
                        <Description xml:lang="en">WORKER9, TEST</Description>
                    </InvoiceDetailItemReference>
                    <SubtotalAmount>
                        <Money currency="USD">-1250.00</Money>
                    </SubtotalAmount>
                    <Tax>
                        <Money currency="USD">0.00</Money>
                        <Description xml:lang="en"/>
                    </Tax>
                    <GrossAmount>
                        <Money currency="USD">-1250.00</Money>
                    </GrossAmount>
                    <NetAmount>
                        <Money currency="USD">-1250.00</Money>
                    </NetAmount>
                    <Distribution>
                        <Accounting name="Accounting">
                           <AccountingSegment id="100">
                               <Name xml:lang="en">PERCENTAGE</Name>
                               <Description xml:lang="en">PERCENTAGE</
Description>
                           </AccountingSegment>
                           <AccountingSegment id="000000100445">
                               <Name xml:lang="en">InternalOrder</Name>
                               <Description xml:lang="en">for Performance Test
908</Description>
                           </AccountingSegment>
                           <AccountingSegment id="400000">
                               <Name xml:lang="en">GeneralLedger</Name>
                               <Description xml:lang="en">ID</Description>
                           </AccountingSegment>
                        </Accounting>
                        <Charge>
                            <Money currency="USD">-1250.00</Money>
                        </Charge>
                    </Distribution>
                    <Comments/>
<Extrinsic name="Ariba.SpendCategory">Contingent Labor</Extrinsic>
                </InvoiceDetailItem>
            </InvoiceDetailOrder>
            <InvoiceDetailSummary>
                <SubtotalAmount>
                    <Money currency="USD">-1250.00</Money>
                </SubtotalAmount>
                <Tax>
                    <Money currency="USD">0.00</Money>
                    <Description xml:lang="en"/>
                </Tax>
                <GrossAmount>
                    <Money currency="USD">-1250.00</Money>
                </GrossAmount>
                <NetAmount>
                    <Money currency="USD">-1250.00</Money>
                </NetAmount>
                <DueAmount>
                    <Money currency="USD">-1250.00</Money>
                </DueAmount>
            </InvoiceDetailSummary>
        </InvoiceDetailRequest>
    </Request>
</cXML>
```

Related Information

cXML Invoice Download Request with System ID as Connector Argument [page 22]

Schema Definition File Descriptions [page 15]

6 Request Status Messages and Return Codes

When downloading or uploading data using the SAP Fieldglass Web Service Technical Specification, you can confirm successful transactions and troubleshoot unsuccessful transactions with these status messages and corresponding return codes.

Status Messages and Return Codes

Status Message	Return Code
SUCCESS/Data Ready	0
No information finds.	-1
Bad transaction ID or reference.	3
Unable to authenticate.	100
Too many records.	104
General Error.	106
No data found.	107
Invalid values for Object ID.	255
Internal Error/Unknown Exception.	500

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.

www.sap.com/contactsap

THE BEST RUN SAP