Translation Web Services Specification 1.0

Working Draft 1.0.2, 12 July 2006

Translation Web Services Specification Identifier

trans-ws-spec-xhtml-1.0 (27/11/05 ABZ)

Location

To Be Added

Chair

Peter Reynolds

Editors

- Peter Reynolds
- Bill Looby
- Stephen Flinter
- Andrzej Zydro
- Kevin Bargary
- Magnus Martikainen

Technical Committee:

Translation Web Services

Abstract

This document defines the Translation Web services specification. The purpose of this Web service is to provide a standard mechanism for the use of Web services within the translation industry.

Status of this Document

This is a draft Specification which the OASIS Translation Web services Technical Committee is working on with a view to approving as a committee specification. Comments may be be submitted to the OASIS Translation Web services Technical Committee at: http://www.oasis-open.org/committees/comments/form.php?wg_abbrev=trans-ws

Notices

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information
on OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification, can be obtained from the OASIS President.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS President.

Copyright © OASIS Open 2005. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself does not modify in any way, such as by removing the copyright notice or references to OASIS, except as needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it into languages other than English. The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an “AS IS” basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Table of Contents

1. Introduction
2. Service Support
   2.1 retrieveServiceList
3. Security
4. Translation and Request Quote
   4.1 requestQuote
   4.2 retrieveQuote
   4.3 acceptQuote
   4.4 submitJob
5. Status, Notification and Delivery
   5.1 retrieveJobInformation
      5.1.1 Job History
   5.2 retrieveJob
   5.3 rejectJob
   5.4 retrieveActiveJobsList
   5.5 retrieveFullJobsList
   5.6 suspendJob
   5.7 resumeJob
   5.8 cancelJob
6. Reference Files
1. Introduction

This document details the draft specification for a standard for using Web services within the translation industry. The document has been drafted by the Translation Web service technical committee at OASIS. It is focused on specifying the calls needed to use Web services for the submission and retrieval of translation and other work.

The methods are grouped into categories in order to simplify the breakdown of discussions. These are described as follows:

2. Service Support

   ● This section describes how a list of services can be made available so that this can be searched.

3. Security

   ● This section describes how security should be dealt with.

4. Translation and Request Quote

   ● This section describes the means of requesting and receiving quotes for a particular localization task and job submission.

5. Status, Notification and Delivery

   ● This section describes how the status and job information can be queried.

6. Reference Files

   ● This section describes how reference files can be associated with a particular job.
2. Service Support

This section covers a single method only, and that is the "Query Support" method, however the decisions made as to the means of service specification, will impact most if not all other method definitions.

The method used for Service support is the `retrieveServiceList` call. The purpose of this call is to get a complete list of services offered by a particular vendor. This will include the languages dealt with and services offered by a particular vendor.

### 2.1 retrieveServiceList

**Purpose**

1. Allow a user to query range of support offered by a particular service implementation.
2. Provide definitions that can be further referenced when requesting quotes/translations (e.g. Service offerings)

**Passed In**

- nothing

**Returned**

- Languages
- Service types
- Mime Types
- Domains

**Support specification**

Note: For the following we should decide whether we wish to include it, and if so, whether its Explicit (and restricted), Free Format or Extensible (i.e. some definitions provided only).

<table>
<thead>
<tr>
<th>Support</th>
<th>Comment</th>
<th>Include, Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Languages support</td>
<td>Language support</td>
<td>Yes, Explicit</td>
</tr>
<tr>
<td>Service Types</td>
<td>See <a href="#">Appendix A</a></td>
<td>Yes, Combination</td>
</tr>
<tr>
<td>Mime Types</td>
<td>Types of file supported Yes, Extensible</td>
<td></td>
</tr>
<tr>
<td>Domain</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Schema**

**Request**

```xml
<xsd:element name="retrieveServiceListRequest">
  <xsd:complexType>
    <xsd:sequence/>
  </xsd:complexType>
</xsd:element>
```
<xsd:element name="retrieveServiceListResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:languagePairs"/>
      <xsd:element ref="types:services"/>
      <xsd:element ref="types:mimeTypes"/>
      <xsd:element ref="types:domains"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="locale">
  <xsd:complexType>
    <xsd:attribute name="country" type="iso3166:CodeContentType_3166" use="required"/>
    <xsd:attribute name="language" type="iso639:CodeContentType_639" use="required"/>
    <xsd:attribute name="variant" type="xsd:string" use="optional"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="source">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:locale"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="target">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:locale"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="languagePair">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:source"/>
      <xsd:element ref="types:target"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="languagePairs">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="1" maxOccurs="unbounded" ref="types:languagePair"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
Sample Message

Request

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
 xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <retrieveServiceListRequest xmlns="urn:oasis:names:tc:wstrans:v1:types"/>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
 xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <retrieveServiceListResponse xmlns="urn:oasis:names:tc:wstrans:v1:types"/>
  </soapenv:Body>
</soapenv:Envelope>
```
3. Security

This specification relies on OASIS WS-Security standard to provide basic security during a web service transaction taking place
between two or more parties. WS-Security provides an end-to-end message level security that achieves 3 goals:

1. To provide message integrity so that the parties involved can guarantee that the message was not modified while in transit through various routers. Tickets or certificates are passed using the XML Signature spec.
2. To provide confidentiality over the message so that the message information cannot be sniffed or read while passing through or in transit. Confidentiality is implemented using XML Encryption spec. Specifically, WS-Security uses three tags: Encrypted Data, Encrypted Key and Reference List.
3. To provide a way to authenticate each party via security tokens such as username/password, kerberos tickets or x.509 certificate. Username/password require pre-knowledge of each other.

The default mechanism which this spec recommends is username/password over SSL.

WS-Security specification provides several methods in which to secure communications. Two systems can conform to the WS-Security spec and still fail to authenticate each other if one system only supports, say, username/password while the other expects digital signatures. Consequently, this specification also recommends WS-Security Policy to specify security policies that define what message integrity it supports, and/or which encryption algorithm it accepts regarding confidentiality.

The recommendation of using username/password over SSL is the minimum level of security. Additional security measures can be implement by agreement between the parties. Future specifications may specify additional security measures.


---

### 4. Translation and Request Quote

This section covers the request for a quote and the request for translations, as by and large, they require the same information to be submitted.

Reject quote has been removed from the previous specification draft as this is probably not core to requirements (and quotes will eventually go out of date regardless)

Suspend/Cancel has been moved to the last section as again more a useful than core requirement.

**Job Identifiers (Job Tickets)**

One definition shared across all methods in this section and in fact in future sections, is job identifiers.

1. Job identifiers are created on the customer side. They can be created before there has been a quote request or job submission. This could be used to submit a resource such as a translation memory prior to the job submission.
2. A unique identifier will be allocated to each individual atomic file/language entity. The granularity of the entity will be defined at the “file/language” pair level as submitted by the customer.
3. The file itself may be an archive file that contains more than one translatable file. No further granularity in identifying the job will be allocated over and above the containing archive file level.
4. The standard will not concern itself with any grouping of the identifiers into any form of ontological entities or otherwise. It is up to the customer and/or supplier to implement any desired grouping of the unique identifiers for their own purposes (i.e. overall deliverable tracking or billing).
5. The standard is not concerned with the format of the identifier.

**Getting a Quote and Placing an Order**
Call 1

**requestQuote**

Job ticket created by publisher. Meta data sent about the job. The location of a file can be indicated using a URI. Quote information returned

Call 2

**retrieveQuote**

Quote is retrieved by the caller

Call 3

**acceptQuote**

Quote is accepted by the publisher. Payment information is sent

As an alternative, the caller can simplify the transaction:

Call 1

**submitJob**

Job ticket created by publisher. Meta data sent about the job. File can be attached as a SOAP attachment. Quote information returned

### 4.1 requestQuote

#### Purpose

This is the point at which meaningful information about the job is needed by the system in order to estimate costs and return them to the user.

#### Format

Two formats (i.e. two methods) should be available, however the first format may not produce a binding quote.

#### Passed In

- Job Ticket
- Count group
  - Count
- Job Meta information
  - Service
  - Domain
  - Language Pair
  - Required by
  - Content information

#### Returned

- Job-ticket
- Available
Schema

Request

```xml
<xsd:element name="requestQuoteRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket"/>
      <xsd:element ref="types:count-group"/>
      <xsd:element ref="types:service"/>
      <xsd:element ref="types:domain"/>
      <xsd:element ref="types:languagePair"/>
      <xsd:element ref="types:requiredBy"/>
      <xsd:element ref="types:info"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="ticket">
  <xsd:complexType>
    <xsd:attribute name="jobId" type="string" use="required"/>
    <xsd:attribute name="projectId" type="string" use="required"/>
    <xsd:attribute name="userId" type="string" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="count-group">
  <xsd:complexType>
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="count" type="xlf:ElemType_count"/>
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="requiredBy">
  <xsd:complexType>
    <xsd:attribute name="date" type="dateTime" use="optional"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="info" type="string"/>
```

Response

```xml
<xsd:element name="requestQuoteResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket"/>
      <xsd:element ref="types:available"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```
Sample Message

Request

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <requestQuoteRequest xmlns="urn:oasis:names:tc:wstrans:v1:types">
      <ticket jobId="job-001" projectId="proj-001" userId="user-001"/>
      <count-group name='project-count'>
        <count count-type="total" unit="words">1001</count>
      </count-group>
    </requestQuoteRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
```
4.2 retrieveQuote

Purpose

This call allows for the retrieval of the quote.

Passed In

- Job-ticket

Returned

- Job-ticket
- Quote URI

A URI is returned when the retrieveQuote call is used. This shows the address of the quote for this particular job.

Schema

Request

```xml
<xsd:element name="retrieveQuoteRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Response

```xml
<xsd:element name="retrieveQuoteResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket"/>
      <xsd:element ref="types:quoteUri"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```
Sample Message

Request

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
 xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <retrieveQuoteRequest xmlns="urn:oasis:names:tc:wstrans:v1:types">
      <ticket jobId="job-001" projectId="proj-001" userId="user-001"/>
    </retrieveQuoteRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
 xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <retrieveQuoteResponse xmlns="urn:oasis:names:tc:wstrans:v1:types">
      <ticket jobId="job-001" projectId="proj-001" userId="user-001"/>
    </retrieveQuoteResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

4.3 acceptQuote

Purpose

Using the Job ticket a quote may be accepted, thereby initiating translation of the job. Payment details in the form of either Purchase Order or Credit Card Info should be supplied.

Passed In

- Job-ticket
- Purchase Order (from recognised supplier)
- Purchase Order URI - This is an optional parameter. A URI could be provide to the address where a copy of the PO can be located.
Returned

- Job-ticket
- Acknowledgement of job initiation

**Schema**

**Request**

```xml
<xsd:element name="acceptQuoteRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket"/>
      <xsd:element ref="types:purchaseOrder"/>
      <xsd:element ref="types:purchaseOrderUri"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="purchaseOrder">
  <xsd:complexType>
    <xsd:attribute name="id" type="string" use="required"/>
  </xsd:complexType>
</xsd:element>
```

**Response**

```xml
<xsd:element name="acceptQuoteResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket"/>
      <xsd:element ref="types:acknowledgement"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="acknowledgement" type="string"/>
```

**Sample Message**

**Request**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <acceptQuoteRequest xmlns="urn:oasis:names:tc:wstrans:v1:types">
      <ticket jobId="job-001" projectId="proj-001" userId="user-001"/>
      <purchaseOrder id="PO-123-456"/>
    </acceptQuoteRequest>
  </soapenv:Body>
</soapenv:Envelope>
```
4.4 submitJob

As an alternative to requestQuote & acceptQuote, where a relationship already exists, or the cost is pre-paid for all required translations, or perhaps the translation service is deployed in-house and is a MT engine, there may be a case to simply call translate as the quote/accept mechanism isn’t required.

Passed In

- Job Ticket
- Count group
  - Count
- Job Meta information
  - Service
  - Domain
  - Language Pair
  - Required by
  - Content information
- Purchase order
- Purchase Order URI - This is an optional parameter. A URI could be provide to the address where a copy of the PO can be located.

Returned

- Job-ticket
- Acknowledgement

Schema
Request

```xml
<xsd:element name="submitJobRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket"/>
      <xsd:element ref="types:count-group"/>
      <xsd:element ref="types:service"/>
      <xsd:element ref="types:domain"/>
      <xsd:element ref="types:languagePair"/>
      <xsd:element ref="types:requiredBy"/>
      <xsd:element ref="types:info"/>
      <xsd:element ref="types:purchaseOrder"/>
      <xsd:element ref="types:purchaseOrderUri"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="ticket">
  <xsd:complexType>
    <xsd:attribute name="jobId" type="string" use="required"/>
    <xsd:attribute name="projectId" type="string" use="required"/>
    <xsd:attribute name="userId" type="string" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="count-group">
  <xsd:complexType>
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="count" type="xlf:ElemType_count"/>
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="requiredBy">
  <xsd:complexType>
    <xsd:attribute name="date" type="dateTime" use="optional"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="info" type="string"/>
```

Response

```xml
<xsd:element name="submitJobResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket"/>
      <xsd:element ref="types:acknowledgement"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```
Sample Message

Request

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <submitJobRequest xmlns="urn:oasis:names:tc:wstrans:v1:types">
      <ticket jobId="job-001" projectId="proj-001" userId="user-001"/>
      <count-group name='project-count'>
        <count count-type="total" unit="words">1001</count>
      </count-group>
      <service>01.00</service>
      <domain>01.00</domain>
      <languagePair>
        <source>
          <locale country="US" language="en"/>
        </source>
        <target>
          <locale country="FR" language="fr"/>
        </target>
      </languagePair>
      <requiredBy date="2004-10-14T09:47:07.415Z"/>
      <info>Info</info>
      <purchaseOrder id="PO-123-456"/>
      <purchaseOrderUri uri="http://www.customer-site.com/trans-ws/download/P01.pdf"/>
    </submitJobRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <submitJobResponse xmlns="urn:oasis:names:tc:wstrans:v1:types">
      <ticket jobId="job-001" projectId="proj-002" userId="user-003"/>
      <acknowledgement>Quantity: 10000 words
        Service: 01.00
        Domain: 01.00
        LanguagePair: US_en -> FR_fr
        RequiredBy: 05-Oct-04
        Info: Info
        PurchaseOrder: PO-123-456
      </acknowledgement>
    </submitJobResponse>
  </soapenv:Body>
</soapenv:Envelope>
```
5. Status, Notification and Delivery

To support management of submitted/accepted jobs the Web Service must provide means for the caller to:

- Get a list of all currently active jobs (retrieveActiveJobsList), or get a list of all jobs whether active or not (retrieveFullJobsList)
- View all available information about a job (retrieveJobInformation)
- Find out the current status of a job, e.g. to determine if it is proceeding according to schedule, if there have been changes to expected delivery dates, or if the job is completed (retrieveJobInformation)
- Download a completed job (retrieveJob)
- Temporarily Suspend a job (suspendJob)
- Resume a temporarily suspended job (resumeJob)
- Cancel a job (cancelJob)

Typical use cases are described in separate sections below.

Notifications

In the initial version of this specification there will be no active notification mechanism. Instead a polling mechanism will need to be used by the caller. The polling mechanism has the advantage that it does not require the caller to provide a means to be contacted explicitly by the Web Service.

It is the caller's responsibility to check the Web Service to find out if things have changed.

Note: Presentation of a return Web Service or WSBPEL notification messages may be used in future versions.

Example: Getting Status Information

A list of active jobs can be retrieved using retrieveActiveJobsList. For any job ticket retrieveJobInformation can be used to get the status information.

Call 1

retrieveActiveJobsList The caller gets a list of all active jobs associated with their account.
This is returned as a list of job tickets

Call 2

retrieveJobInformation The caller gets status information for each job.
The current job status can be deducted from the history section of the response. Here is an example of what this section could look like:

```
<history>
  <event stage="job" status="start" timeStamp="2004-12-01T18:00:00.000Z">
  <event stage="evaluation" status="start" timeStamp="2004-12-01T18:00:00.000Z">
```
From this example we can see that the job has completed evaluation stage and is currently in pre-translation, and that the estimated completion date for the whole job has slipped to 2005-02-08.

More information about the job history can be found in the Job History section.

Example: Downloading All Completed Jobs

At any time the list of active jobs can be retrieved using `retrieveActiveJobsList`. Then for each job ticket `retrieveJobInformation` can be used to get the status information. The caller can then check whether a job is complete and if so use `retrieveJob` to get the document returned.

Call 1

`retrieveActiveJobsList` The caller gets a list of all active jobs associated with their account. This is returned as a list of job tickets.

Call 2

`retrieveJobInformation` For each Job the caller can get the status information. If the history in the response has an event where `stage="job"` and `status="complete"` then the job is completed and can be downloaded.

Call 3

`retrieveJob` The caller downloads the completed job.

5.1 `retrieveJobInformation`

This call allows the caller to view the Job history for any particular job. This might be used together with `retrieveActiveJobsList` to check the status of all jobs for the caller and to download them when the status changes to complete. The status information uses the event element. This gives the status for each event that takes place when a job is being worked on.

Passed In

1. Job-ticket

Returned

1. Job Ticket
2. Count group
   1. Count
3. Job Meta information
   1. Service
   2. Domain
3. Language Pair
4. Required by
5. Content information

4. Purchase order
5. Purchase Order URI - This is an optional parameter. A URI could be provide to the address where a copy of the PO can be located.
6. Associations
7. History
   1. Event
   2. Stage
   3. Status
   4. Calendar
   5. Time stamp
   6. Additional Information (optional)

History

The job history is described in detail in the Job History section.

Here is an example:

```
<history>
  <event stage="job" status="start" timeStamp="2004-01-01T18:00:00.000Z">
    <event stage="evaluation" status="start" timeStamp="2004-01-01T18:00:00.000Z">
      <event stage="job" status="estimated completion" calendar="2004-02-02T18:00:00.000Z" timeStamp="2004-01-01T18:00:00.000Z">
        <event stage="evaluation" status="complete" timeStamp="2004-01-03T18:00:00.000Z">
          <event stage="job" status="estimated completion" calendar="2004-02-04T18:00:00.000Z" timeStamp="2004-01-06T18:00:00.000Z">
            </event>
      </event>
    </event>
  </event>
</history>
```

Schema

Request

```
<xsd:element name="retrieveJobInformationRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Response

```
<xsd:element name="retrieveJobInformationResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```
<xsd:element name="count-group">
  <xsd:complexType>
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="count" type="xlf:ElemType_count"/>
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="associations">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" ref="types:resource"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="resource">
  <xsd:complexType>
    <xsd:attribute name="uri" type="anyURI" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="history">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" ref="types:event"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="event">
  <xsd:complexType>
    <xsd:attribute name="timeStamp" type="xsd:dateTime" use="required"/>
    <xsd:attribute name="stage" type="xsd:string" use="required"/>
    <xsd:attribute name="status" type="TransWSStatus:CodeContentType_status" use="required"/>
    <xsd:attribute name="date" type="xsd:date" use="optional"/>
    <xsd:attribute name="additionalInformation" type="xsd:string" use="optional"/>
  </xsd:complexType>
</xsd:element>
Sample Message

Request

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <retrieveJobInformationRequest xmlns="urn:oasis:names:tc:wstrans:v1:types">
      <ticket jobId="job-001" projectId="proj-002" userId="user-003"/>
    </retrieveJobInformationRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <retrieveJobInformationResponse xmlns="urn:oasis:names:tc:wstrans:v1:types">
      <ticket jobId="job-001" projectId="proj-002" userId="user-003"/>
      <count-group name='project-count'>
        <count count-type="total" unit="words">1001</count>
      </count-group>
      <service name="01.00"/>
      <domain>01.00</domain>
      <languagePair>
        <source>
          <locale country="US" language="en"/>
        </source>
        <target>
          <locale country="FR" language="fr"/>
        </target>
      </languagePair>
      <requiredBy date="2004-12-01T18:00:00.000Z"/>
      <info>Info string</info>
      <purchaseOrder id="PO-123-456"/>
      <associations>
        <resource uri="http://www.simship.com/TM.zip"/>
      </associations>
    </retrieveJobInformationResponse>
  </soapenv:Body>
</soapenv:Envelope>
```
5.1.1 Job History

This topic explains the job history that can be retrieved from `retrieveJobInformation` in more details.

Overview

The job history is comprised of a list of chronological events. Each event is represented by an `<event>` element with the following attributes:

- `stage` – (required) identifies what the event applies to, e.g. a particular task or the entire job.
- `status` – (required) the status value that the event represents. This is always interpreted in the scope of the `stage` value.
- `date` – (optional) a date value for status values that are associated with dates, such as the estimated completion date or the start date of a stage.
- `additionalInformation` – (optional) any additional information that is associated with the event, if applicable. For example the reason specified why a job was rejected through a call to `rejectJob`.
- `timeStamp` – (required) the time and date that the event was recorded in the history.

The different values and how they work together is in the following sections.

Event Stage Values

The following table explains the possible values for the `stage` attribute of a history event.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>job</td>
<td>Overall job status</td>
</tr>
<tr>
<td>quotation</td>
<td>Quotation in progress, the quote is being created or has been created but has not yet been accepted.</td>
</tr>
<tr>
<td>evaluation</td>
<td>Evaluation of the job</td>
</tr>
<tr>
<td>pre-translation</td>
<td>Preparatory work before actual translation work commences</td>
</tr>
<tr>
<td>translation</td>
<td>Translation work and directly related activities</td>
</tr>
<tr>
<td>post-translation</td>
<td>Work after the translation stage has completed</td>
</tr>
<tr>
<td>engineering</td>
<td>Localization engineering work</td>
</tr>
<tr>
<td>internationalization</td>
<td>Internationalization work</td>
</tr>
<tr>
<td>quality assurance</td>
<td>Quality Assurance work</td>
</tr>
<tr>
<td>miscellaneous</td>
<td>General/unspecific localization work</td>
</tr>
<tr>
<td>(other)</td>
<td>Application defined (custom) work stage and/or status scope.</td>
</tr>
</tbody>
</table>

Event Status Values

The following table explains the possible values for the `status` attribute of a history event.
<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>Work stage started at the time indicated by the timeStamp value.</td>
</tr>
<tr>
<td>complete</td>
<td>Work stage completed at the time indicated by the timeStamp value.</td>
</tr>
<tr>
<td>cancelled</td>
<td>Work stage cancelled at the time indicated by the timeStamp value.</td>
</tr>
<tr>
<td>on hold</td>
<td>The work task was put on hold by the Web Service Host at the time indicated by the timeStamp value.</td>
</tr>
<tr>
<td>suspended</td>
<td>The job was suspended by the Web Service caller through a call to suspendJob at the time indicated by the timeStamp value.</td>
</tr>
<tr>
<td></td>
<td>This status value can currently only be used with stage=&quot;job&quot;.</td>
</tr>
<tr>
<td>resumed</td>
<td>A suspended job was resumed by the Web Service caller through a call to resumeJob at the time indicated by the timeStamp value.</td>
</tr>
<tr>
<td></td>
<td>This status value can currently only be used with stage=&quot;job&quot;.</td>
</tr>
<tr>
<td>resource associated</td>
<td>A resource was associated with the job through a call to associateResource. The timestamp value indicates when the actual association was made, which could in fact be before the job was created. The additional information attribute contains the URI of the associated resource.</td>
</tr>
<tr>
<td></td>
<td>This value can only be used with stage=&quot;job&quot;.</td>
</tr>
<tr>
<td>resource disassociated</td>
<td>A resource was detached from the job through a call to disassociateResource. The timestamp value indicates when the association between the resource and the job was terminated. The additional information attribute contains the URI of the disassociated resource.</td>
</tr>
<tr>
<td></td>
<td>This value can only be used with stage=&quot;job&quot;.</td>
</tr>
<tr>
<td>estimated completion</td>
<td>An estimated completion date for the work task was set by the Web Service Host. The estimated completion date is indicated by the date value. The timeStamp value indicates the time this estimation was published.</td>
</tr>
<tr>
<td>retrieved</td>
<td>The completed job was downloaded by a call to retrieveJob at the time indicated by the timeStamp value.</td>
</tr>
<tr>
<td></td>
<td>This value can only be used with stage=&quot;job&quot;.</td>
</tr>
<tr>
<td>required by</td>
<td>A new deadline was specified for the job. The new required completion date is indicated by the date value. The deadline update was recorded at the time indicated by the timeStamp value.</td>
</tr>
<tr>
<td></td>
<td>This value can only be used with stage=&quot;job&quot;.</td>
</tr>
<tr>
<td>rejected</td>
<td>The completed job was rejected by the Web Service caller through a call to rejectJob at the time indicated by the timeStamp value. The information about why the job was rejected is available through the additional information attribute.</td>
</tr>
<tr>
<td></td>
<td>This value can only be used with stage=&quot;job&quot;.</td>
</tr>
<tr>
<td>x-(other)</td>
<td>Application defined (custom) status value or a free form comment in clear text, prefixed by x- to make it easy to distinguish from the values defined in this specification.</td>
</tr>
</tbody>
</table>

There is no way for the caller to explicitly set a new “Required By” date. This can only be achieved through communication not covered by this version of the standard. This will be addressed in a future version of the specification.

The Past Never Changes

Just like in real life the past, here in the shape of already recorded events, should never change during the lifetime of a job and beyond.

As the job progresses or properties of the job changes new events will be added after the already existing events in the history. It should always be possible to get an overview of the entire job history, including all events relevant for the job, by looking at the history. This can be important in order to find out what really happened during a job at a later stage.
The job history can also be important as a means of communicating changes in expected delivery dates. Below is an example that illustrates this.

### How to find out if a Job has Slipped

The current job status can be deducted from the job history. In this example it looks like this:

```xml
<history>
  <event stage="job" status="start" time=2004-12-01T18:00:00.000Z">
    <event stage="evaluation" status="start" time=2004-12-01T18:00:00.000Z">
      <event stage="job" status="estimated completion" date=2005-02-01T18:00:00.000Z" time=2004-12-01T18:00:00.000Z">
        <event stage="evaluation" status="complete" time=2005-01-05T13:34:10.000Z">
          <event stage="pre-translation" status="start" time=2005-01-05T13:34:10.100Z">
            <event stage="job" status="estimated completion" date=2005-02-08T18:00:00.000Z" time=2005-01-06T18:32:00.000Z">
            </event>
        </event>
      </event>
    </event>
  </event>
</history>
```

The original estimated completion date for the job stage (i.e. the whole job) was 2005-02-01, as can be seen from the third event in the list. The timestamp further indicates that this original estimate was made on 2004-12-01.

The last event in the list contains a second estimated completion date for the job, which indicates that the completion of the job is expected to slip to 2005-02-08. The timestamp indicates that this updated estimate was recorded on 2005-01-06.

The order of events in the history should be chronological, so by searching for the last occurrence of the estimated completion date for the stage with value job you should always be able to find the most recent estimated date of completion.

Note that estimated completion date can also be used for other stage values, in which they simply provide an indication on when that particular stage of the job is estimated to be completed.

### 5.2 retrieveJob

This call is a mechanism for downloading a completed job. It can be used in conjunction with the retrieveJobInformation call. When the job status has been changed to complete this call can be used to retrieve the URI from which to download the file.

**Passed In**

- Job-ticket

**Returned**

- **Job-ticket** The job ticket.
- **Resource URI** URI that can be used to download the file containing the completed job.
Rejection deadline

If the job has not been rejected (by calling `rejectJob`) before this deadline is reached it will be deemed accepted, and the job will no longer be considered active.

Schema

Request

```xml
<xsd:element name="retrieveJobRequest">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="types:ticket"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

Response

```xml
<xsd:element name="retrieveJobResponse">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="types:ticket"/>
            <xsd:element ref="types:resource" />
            <xsd:element ref="types:rejectionDeadline"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

Sample Message

Request

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Body>
        <retrieveJobRequest xmlns="urn:oasis:names:tc:wstrans:v1:types">
            <ticket jobId="job-001" projectId="proj-002" userId="user-003"/>
        </retrieveJobRequest>
    </soapenv:Body>
</soapenv:Envelope>
```
Response

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <retrieveJobResponse xmlns="urn:oasis:names:tc:wstrans:v1:types">
      <ticket jobId="job-001" projectId="proj-002" userId="user-003"/>
      <rejectionDeadline date="2004-12-31T23:59:59.389Z"/>
    </retrieveJobResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

See also: Appendix D, retrieveActiveJobsList, retrieveJobInformation, submitJob, rejectJob, acceptQuote

---

5.3 rejectJob

This call is allows for the translation to be rejected if it is incomplete or requires correction. It also allows for feedback to be submitted.

A rejected job will remain active even after the rejection deadline returned from retrieveJob.

An entry with the following information will be added to the <history> element returned from retrieveJobInformation:

```xml
<event stage="job" status="rejected" timeStamp="2005-01-01T12:00:00.000Z"
  additionalInformation="Rejection Details"/>
```

Passed In

- Job ticket
- Rejection details
- Content information

Returned

- Job ticket
- Acknowledgement

Schema

Request
Sample Message

Request

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <rejectJobRequest xmlns="urn:oasis:names:tc:wstrans:v1:types">
      <ticket jobId="job-001" projectId="proj-002" userId="user-003"/>
      <reason>Poor quality translation. Contact PM for details.</reason>
      <info>These are the files which were translated into French and German.</info>
    </rejectJobRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <rejectJobResponse xmlns="urn:oasis:names:tc:wstrans:v1:types">
      <ticket jobId="job-001" projectId="proj-002" userId="user-003"/>
    </rejectJobResponse>
  </soapenv:Body>
</soapenv:Envelope>
```
5.4 retrieveActiveJobsList

This call allows the caller to retrieve a list of their currently active jobs. This can be used together with retrieveJobInformation to get the current status for the active jobs.

Passed In

- Nothing

Returned

- A list of job tickets representing the currently active jobs

Schema

Request

```xml
<xsd:element name="retrieveActiveJobsListRequest">
  <xsd:complexType>
    <xsd:sequence/>
  </xsd:complexType>
</xsd:element>
```

Response

```xml
<xsd:element name="retrieveActiveJobListResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" ref="types:ticket"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Sample Message

Request

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
</soapenv:Envelope>
```
5.5 retrieveFullJobsList

This is similar to the retrieveActiveJobsList but retrieves a list of all jobs associated with a particular caller, whether or not they are active.

Passed In

- Nothing

Returned

- A list of job tickets representing all active and not active jobs for the caller

Schema

Request

```xml
<xsd:element name="retrieveFullJobsListRequest">
  <xsd:complexType>
    <xsd:sequence/>
  </xsd:complexType>
</xsd:element>
```
Response

```xml
<xsd:element name="retrieveFullJobListResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" ref="types:ticket"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Sample Message

Request

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <retrieveFullJobsListRequest xmlns="urn:oasis:names:tc:wstrans:v1:types"/>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <retrieveFullJobsListResponse xmlns="urn:oasis:names:tc:wstrans:v1:types">
      <ticket jobId="job-001" projectId="project-001" userId="user-001"/>
      <ticket jobId="job-002" projectId="project-002" userId="user-002"/>
      <ticket jobId="job-003" projectId="project-003" userId="user-003"/>
    </retrieveFullJobsListResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

See also: `retrieveActiveJobsList`

5.6 suspendJob

The caller may request that any currently active, not yet completed job be temporarily suspended by calling this method.

Note: This is a request only as the service cannot guarantee to comply with this immediately. There may still be a cost incurred.
A suspended job can be cancelled (by calling `cancelJob`) or resumed at a later stage (by calling `resumeJob`).

`suspendJob` will fail if the job status is complete.

An entry with the following information will be added to the history returned from `retrieveJobInformation`:

```xml
<event stage="job" status="suspended" timeStamp="2005-01-01T12:00:00.000Z"/>
```

**Passed In**

- Job-ticket

**Returned**

- Job ticket
- Success indicator
- Additional information

**Schema**

**Request**

```xml
<xsd:element name="suspendJobRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

**Response**

```xml
<xsd:element name="suspendJobResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket"/>
      <xsd:element ref="types:succeeded"/>
      <xsd:element ref="types:info"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

**Sample Message**

**Request**

```xml
<?xml version="1.0" encoding="UTF-8"?>
```
Response

5.7 resumeJob

Using this method the caller may request that a job which has previously been temporarily suspended (by a call to suspendJob) is resumed.

Note: This is a request only as the service cannot guarantee to comply with this immediately. There may still be a cost incurred.

An entry of the following format will be added to the <history> element returned from retrieveJobInformation:

```xml
<event stage="job" status="resumed" timeStamp="2005-01-01T12:00:00.000Z"/>
```

Passed In

- Job-ticket

Returned

- Job ticket
- Success indicator
- Additional information
Schema

Request

```xml
<xsd:element name="resumeJobRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Response

```xml
<xsd:element name="resumeJobResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket"/>
      <xsd:element ref="types:succeeded"/>
      <xsd:element ref="types:info"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Sample Message

Request

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
 xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <resumeJobRequest xmlns="urn:oasis:names:tc:wstrans:v1:types">
      <ticket jobId="job-001" projectId="proj-002" userId="user-003"/>
    </resumeJobRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
 xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
```

5.8 cancelJob

Using this method the caller may request that any currently active, not completed job be cancelled.

Note: This is a request only as the service cannot guarantee to comply with this immediately. There may still be a cost incurred.

cancelJob will fail if the job is in a completed state.

An entry with the following information will be added to the <history> element returned from retrieveJobInformation:

```
<event stage="job" status="cancelled" timeStamp="2005-01-01T12:00:00.000Z"/>
```

Passed In

- Job-ticket

Returned

- Job ticket
- Success indicator
- Additional information

Schema

Request

```
<xsd:element name="cancelJobRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Response

```
<xsd:element name="cancelJobResponse">
```
Sample Message

Request

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <cancelJobRequest xmlns="urn:oasis:names:tc:wstrans:v1:types">
      <ticket jobId="job-001" projectId="proj-002" userId="user-003"/>
    </cancelJobRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <cancelJobResponse xmlns="urn:oasis:names:tc:wstrans:v1:types">
      <ticket jobId="job-001" projectId="proj-002" userId="user-003"/>
      <succeeded>1</succeeded>
      <info>Cancellation succeeded</info>
    </cancelJobResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

6. Reference Files

Both translatable and non-translatable files may be needed for a job.

Examples of non-translatable files include:
The calls described in this section allow for such resource files, as well as other translatable files, to be associated with a job. Some of these resource types, such as translation memories and terminology databases, may be updated as part of executing the job. It is important that the updated resources can be accessed both during the job and as part of the delivery when the job is completed.

In some cases it may even be necessary to upload such resources before the job itself is submitted, e.g. to provide all input required to get an accurate quote or estimate for the job.

For this reason it is possible to call `associateResource` with a job ticket that is not yet known to the system (e.g before `submitJob` has been called).

Resources can be associated with more than one job.

- To find out which resources are associated with a job, call `retrieveJobInformation`.
- To find out which jobs a resource is associated with, call `retrieveResourceInformation`.
- To break the association between a job and a resource, call `disassociateResource`.
- To get a list of all resources, call `retrieveFullResourceList`.

**Example: Job with Translation Memory**

**Call 1**

`associateResource`  
The content owner associates a translation memory with a new job that has not yet been uploaded. A resource URI is returned, which can be used to reference or retrieve the translation memory.

**Call 2**

`requestQuote`  
The content owner submits the rest of the job information and initiates the quoting process. The same job ticket as was passed to the previous call is used here.

**Call 3**

`retrieveQuote`  
The content owner retrieves the quote for the job.

**Call 4**

`acceptQuote`  
The content owner accepts the quote and the job is started.

**Call 5**

`retrieveJob`  
Once the job is completed the content owner retrieves the URI from which the results can be downloaded.

The final version of the translation memory can now be downloaded from the URI returned from the original `associateResource` call.

**See also:** `associateResource`, `disassociateResource`, `retrieveResourceInformation`, `retrieveJobInformation`, `retrieveFullResourceList`
6.1 associateResource

This call allows for the association of a resource file such as translation memory or the URI for a glossary with a particular job. The same resource file can be associated with any number of jobs.

Passed In

- **Job ticket** – The job the resource should be associated with
- **Resource URI** – URI for the resource to be associated with the specified jobs
- **Purpose** – Explains what the resource is intended to be used for

Returned

- **Job ticket**
- **Success Indication**
- **Resource URI** – This may be a different URI from the one passed in.

If the web service implementation returns a different URI from the one passed in this may refer to a modified version of the resource. An example of this would be a translation memory that is updated as part of executing the job.

It is recommended that the passed-in URI should remain active and accessible for the duration of all jobs that the resource has been associated with.

If the resource is associated with more than one jobs, this call should be made for each job.

An entry of the following format will be added to the `<history>` element returned from retrieveJobInformation:

```
<event stage="job" status="resource associated" timeStamp="2005-01-01T12:00:00.000Z" additionalInformation="http://aCompany.com/Resource.tm"/>
```

You can call retrieveResourceInformation to find out which jobs a particular resource URI is associated with.

To remove the association between a resource and a job, call disassociateResource.

Schema

**Request**

```xml
<xsd:element name="associateResourceRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket"/>
      <xsd:element ref="types:resource"/>
      <xsd:element ref="types:purpose"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```
Response

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <associateResourceResponse xmlns="urn:oasis:names:tc:wstrans:v1:types">
      <ticket jobId="job-001" projectId="proj-002" userId="user-003"/>
      <succeeded>0</succeeded>
    </associateResourceResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample Message

Request

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <associateResourceRequest xmlns="urn:oasis:names:tc:wstrans:v1:types">
      <ticket jobId="job-001" projectId="proj-002" userId="user-003"/>
      <purpose usage="reference"/>
    </associateResourceRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <associateResourceResponse xmlns="urn:oasis:names:tc:wstrans:v1:types">
      <ticket jobId="job-001" projectId="proj-002" userId="user-003"/>
      <succeeded>0</succeeded>
    </associateResourceResponse>
  </soapenv:Body>
</soapenv:Envelope>
```
6.2 disassociateResource

Removes the association between a resource and specified jobs.

Passed In

- Resource URI – This can be either the original resource URI passed to associateResource or the URI returned from that call.
- Job ticket – The job which should no longer be associated with the resource.

Returned

- Job ticket
- Success Indication

If the resource is to be removed from more than one jobs, this call should be made for each job.

An entry of the following format will be added to the <history> element returned from retrieveJobInformation:

```xml
<event stage="job" status="resource disassociated" timeStamp="2005-01-01T12:00:00.000Z" additionalInformation="http://aCompany.com/Resource.tm"/>
```

Schema

Request

```xml
<xsd:element name="disassociateResourceRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket"/>
      <xsd:element ref="types:resource"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Response

```xml
<xsd:element name="disassociateResourceResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket"/>
      <xsd:element ref="types:succeeded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```
Sample Message
Request

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <soapenv:Body>
  <disassociateResourceRequest xmlns="urn:oasis:names:tc:wstrans:v1:types">
   <ticket jobId="job-001" projectId="proj-002" userId="user-003"/>
  </disassociateResourceRequest>
 </soapenv:Body>
</soapenv:Envelope>

Response

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <soapenv:Body>
  <disassociateResourceResponse xmlns="urn:oasis:names:tc:wstrans:v1:types">
   <ticket jobId="job-001" projectId="proj-002" userId="user-003"/>
   <succeeded>1</succeeded>
  </disassociateResourceResponse>
 </soapenv:Body>
</soapenv:Envelope>

See also: associateResource, retrieveResourceInformation

6.4 retrieveResourceInformation

Returns all known information about a resource.

Passed In

- URI – This can either be the URI originally supplied with the associateResource call, or the URI returned from it (if different).

Returned
A list of the following:

- **Job ticket** – Job that this resource is currently associated with.
- **Purpose** – The purpose specified for the resource when it was associated with the job in the `associateResource` call.
- **Original URI** – The URI that was passed in to the `associateResource` call.
- **Current URI** – The URI that was returned from the `associateResource` call.

Optional free form information

Note that the same resource URI can be associated with different jobs for potentially different purposes (using separate calls to `associateResource`). This call returns the information about the resource for each job it is associated with separately.

**Schema**

**Request**

```xml
<xsd:element name="retrieveResourceInformationRequest">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="types:resource"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

**Response**

```xml
<xsd:element name="retrieveResource Information Response">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="types:ticket"/>
            <xsd:element ref="types:purpose"/>
            <xsd:element ref="types:getOriginalResource"/>
            <xsd:element ref="types:resource"/>
            <xsd:element ref="types:info"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

**Sample Message**

**Request**

```xml
<?xml version="1.0" encoding="UTF-8"?>
```
6.4 retrieveFullResourceList

Retrieves a list of all resources associated with a particular caller, whether or not they are active.

Passed In

- Nothing

Returned

- A list of resource URIs representing all resources associated with the caller.

Schema

Request

```xml
<xsd:element name="retrieveFullResourceListRequest">
```
Response

```xml
<xsd:element name="retrieveFullResourceListResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" ref="types:resource"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Sample Message

Request

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <retrieveFullResourceListResponse xmlns="urn:oasis:names:tc:wstrans:v1:types">
      <resource uri="http://company.com/server/resource.zip"/>
      <resource uri="file:C:\Temp\File.zip"/>
    </retrieveFullResourceListResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

See also: [retrieveResourceInformation](#)

7. uploadFile
Purpose

This method is optional and should be used to upload assets to the localization suppliers. A URI with the location of this asset is returned and this is used for the other calls in the specification.

Note

The asset should be attached using SOAP messages. It is the responsibility of those using this call to decide which technology they would like to use for attaching the asset.

Passed In

- nothing

Returned

- uri

Schema

Request

```xml
<xsd:element name="fileUpload">
  <xsd:complexType>
    <xsd:sequence/>
  </xsd:complexType>
</xsd:element>
```

Response

```xml
<xsd:element name="fileUploadResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:uri"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

SOAP Attachment

Example of schema for using SOAP attachments

```xml
<binding>
  <operation>
    <input>
      <mime:multipartRelated>
        <mime:part>
```

Appendix A: List of calls and errors

<table>
<thead>
<tr>
<th>Call</th>
<th>Option</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>retrieveServiceList</td>
<td>2</td>
<td>Operation failed</td>
</tr>
<tr>
<td>requestQuote</td>
<td>2</td>
<td>Operation failed, Invalid ticket, un-supported language pairs</td>
</tr>
<tr>
<td>acceptQuote</td>
<td>2</td>
<td>Operation failed, Invalid ticket</td>
</tr>
<tr>
<td>retrieveQuote</td>
<td>2</td>
<td>Operation failed, Invalid ticket</td>
</tr>
<tr>
<td>submitJob</td>
<td>0</td>
<td>Operation failed, Invalid ticket, un-supported language pairs</td>
</tr>
<tr>
<td>retrieveJobInformation</td>
<td>0</td>
<td>Operation failed, Invalid ticket</td>
</tr>
<tr>
<td>retrieveJob</td>
<td>0</td>
<td>Operation failed, Invalid ticket</td>
</tr>
<tr>
<td>rejectJob</td>
<td>1</td>
<td>Operation failed, Invalid ticket</td>
</tr>
<tr>
<td>retrieveActiveJobsList</td>
<td>0</td>
<td>Operation failed, Invalid ticket</td>
</tr>
<tr>
<td>retrieveFullJobsList</td>
<td>2</td>
<td>Operation failed, Invalid ticket</td>
</tr>
<tr>
<td>suspendJob</td>
<td>0</td>
<td>Operation failed, Invalid ticket</td>
</tr>
<tr>
<td>resumeJob</td>
<td>0</td>
<td>Operation failed, Invalid ticket</td>
</tr>
<tr>
<td>cancelJob</td>
<td>0</td>
<td>Operation failed, Invalid ticket</td>
</tr>
<tr>
<td>Operation</td>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------</td>
<td></td>
</tr>
<tr>
<td>associateResource</td>
<td>1 Operation failed, Invalid ticket, Malformed URI</td>
<td></td>
</tr>
<tr>
<td>disassociateResource</td>
<td>1 Operation failed, Invalid ticket, Malformed URI</td>
<td></td>
</tr>
<tr>
<td>retrieveResourceInformation</td>
<td>1 Operation failed, Malformed URI</td>
<td></td>
</tr>
</tbody>
</table>

Option: 0 - must have
1 - should have
2 - may have

## Appendix B: Code Lists

The model for schema enabled presentation and validation of code list for country code, languages and content domains is based on the OASIS UBL recommendation [Universal Business Language (UBL) Code List Rules](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=UBL).

The UBL Code List Rules recommend the “Multiple Namespace Types Method” whereby the elements that represent a code from a particular list are bound to types that may have come from an external organization’s schema module.

The instance of each code instance is related by namespace to the validating schema document e.g.:

```xml
<doc xmlns:iso3166="http://example.iso.org/schemas/CountryCodesISO3166.xsd">
  <langCode>
    <iso3166:CodeContentType_3166>US</iso3166:CodeContentType_3166>
  </langCode>
</doc>
```

The Translation Web Services specification defines the following code verification schemas:

1. [Country Codes ISO3166](http://example.iso.org/schemas/CountryCodesISO3166.xsd) for ISO 3166 Country codes.
2. Language Codes ISO639 for ISO 639.2 language codes.
3. Trans WSDomains for translation specific domain classifications based on ATA (American Translators Association) list of domains.
4. Mime Types for MIME types used in attachments.
5. Trans Scope for the different scopes of the various status messages for a job.
6. Service Types for the different localisation services that may be offered by a translation/localisation vendor.
7. Trans Status for the different statuses a job may be in.

These schemas will be hosted by TBA and can be referenced by the following URL addresses: TBA

## Appendix C: File upload/download

The specifications allow for the use of a method, uploadFile, for the purpose of sending files. This method is optional and should be used to upload assets to the localization suppliers. A URI with the location of this asset is returned and this is used for the other calls in the specification.

You can find details of the uploadFile by [clicking here](#).
Within our committee we have had a number of discussions on how best to upload files. Detailed discussions established that using WWW URI mechanism to specify the location of a file would be a very good way of locating files. This leaves the retrieval mechanism including any security arrangements and clean up to the party responsible for the supply of the file.

The reasoning behind this decision is as follows:

1. SOAP with Attachments is not ideal for very large attachments and we cannot impose any limits on the size of attachments. In certain type of localization task very large files may be exchanged.
2. A common mechanism that is not encumbered by file size, but uses standard W3C location definition is required for all data objects.
3. The supplier of the file may need to implement specific security mechanism that are not covered by SwA.

The URI mechanism provides a common way for specifying the location of all data objects.

---

**Appendix D: Trans WS Sample Service Description WSDL**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="urn:oasis:names:tc:wstrans:v1:service"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:service="urn:oasis:names:tc:wstrans:v1:service"
    xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:bind="urn:oasis:names:tc:wstrans:v1:bind"
    xmlns="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:service name="TransWSService">
        <wsdl:port name="TransWSRecommended_Port" binding="bind:TransWSRecommended_Binding">
            <wsdlsoap:address location="http://www.electonline.org:8080/axis/services/TransWSRecommended_Port"/>
        </wsdl:port>
        <wsdl:port name="TransWSRequired_Port" binding="bind:TransWSRequired_Binding">
            <wsdlsoap:address location="http://www.electonline.org:8080/axis/services/TransWSRequired_Port"/>
        </wsdl:port>
        <wsdl:port name="TransWSOptional_Port" binding="bind:TransWSOptional_Binding">
            <wsdlsoap:address location="http://www.electonline.org:8080/axis/services/TransWSOptional_Port"/>
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

---

**Appendix E: Trans WS Bindings WSDL**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="urn:oasis:names:tc:wstrans:v1:bind"
```
<wsdl:binding name="TransWSRecommended_Binding" type="intf:TransWSRecommended_PT">

<wsdl:operation name="rejectJob">
<wsdl:input name="rejectJobRequest">
<wsdlsoap:body use="literal"/>
</wsdl:input>
<wsdl:output name="rejectJobResponse">
<wsdlsoap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="associateResource">
<wsdl:input name="associateResourceRequest">
<wsdlsoap:body use="literal"/>
</wsdl:input>
<wsdl:output name="associateResourceResponse">
<wsdlsoap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="disassociateResource">
<wsdl:input name="disassociateResourceRequest">
<wsdlsoap:body use="literal"/>
</wsdl:input>
<wsdl:output name="disassociateResourceResponse">
<wsdlsoap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="retrieveResourceInformation">
<wsdl:input name="retrieveResourceInformationRequest">
<wsdlsoap:body use="literal"/>
</wsdl:input>
<wsdl:output name="retrieveResourceInformationResponse">
<wsdlsoap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="retrieveFullResourceList">
<wsdl:input name="retrieveFullResourceListRequest">
<wsdlsoap:body use="literal"/>
</wsdl:input>
<wsdl:output name="retrieveFullResourceListResponse">
<wsdlsoap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="retrieveFullResourceList">
    <wsdl:input name="retrieveFullResourceListRequest">
        <wsdlsoap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="retrieveFullResourceListResponse">
        <wsdlsoap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<wsdl:operation name="UploadFile">
    <wsdl:input name="UploadFileRequest">
        <wsdlsoap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="UploadFileResponse">
        <wsdlsoap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<wsdl:operation name="retrieveJobListForResource">
    <wsdl:input name="retrieveJobListForResourceRequest">
        <wsdlsoap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="retrieveJobListForResourceResponse">
        <wsdlsoap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<wsdl:operation name="submitJob">
    <wsdl:input name="submitJobRequest">
        <wsdlsoap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="submitJobResponse">
        <wsdlsoap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<wsdl:operation name="retrieveJobInformation">
    <wsdl:input name="retrieveJobInformationRequest">
        <wsdlsoap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="retrieveJobInformationResponse">
        <wsdlsoap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<wsdl:operation name="retrieveJob">
    <wsdl:input name="retrieveJobRequest">
        <wsdlsoap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="retrieveJobResponse">
        <wsdlsoap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="retrieveActiveJobsList">
  <wsdl:input name="retrieveActiveJobsListRequest">
    <wsdlsoap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="retrieveActiveJobsListResponse">
    <wsdlsoap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="suspendJob">
  <wsdl:input name="suspendJobRequest">
    <wsdlsoap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="suspendJobResponse">
    <wsdlsoap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="resumeJob">
  <wsdl:input name="resumeJobRequest">
    <wsdlsoap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="resumeJobResponse">
    <wsdlsoap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="cancelJob">
  <wsdl:input name="cancelJobRequest">
    <wsdlsoap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="cancelJobResponse">
    <wsdlsoap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="retrieveServiceList">
  <wsdl:input name="retrieveServiceListRequest">
    <wsdlsoap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="retrieveServiceListResponse">
    <wsdlsoap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
Appendix F: Trans WS Interfaces WSDL

```xml
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="urn:oasis:names:tc:wstrans:v1:intf"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:intf="urn:oasis:names:tc:wstrans:v1:intf"
    xmlns:types="TransWSTypes.xsd">
  <wsdl:types>
    <xsd:schema elementFormDefault="qualified" xmlns="http://www.w3.org/2001/XMLSchema"/>
  </wsdl:types>
</wsdl:definitions>
```
XMLSchema>
<xsd:import namespace="urn:oasis:names:tc:wstrans:v1:types"
schemaLocation="TransWSTypes.xsd"/>
</xsd:schema>

</wsdl:types>
<wsdl:message name="retrieveJobListForResourceResponse">
  <wsdl:part name="parameters" element="types:retrieveJobListForResourceResponse"/>
</wsdl:message>
<wsdl:message name="retrieveJobListForResourceRequest">
  <wsdl:part name="parameters" element="types:retrieveJobListForResourceRequest"/>
</wsdl:message>
<wsdl:message name="disassociateResourceRequest">
  <wsdl:part name="parameters" element="types:disassociateResourceRequest"/>
</wsdl:message>
<wsdl:message name="retrieveResourceInformationRequest">
  <wsdl:part name="parameters" element="types:retrieveResourceInformationRequest"/>
</wsdl:message>
<wsdl:message name="retrieveFullResourceListRequest">
  <wsdl:part name="parameters" element="types:retrieveFullResourceListRequest"/>
</wsdl:message>
<wsdl:message name="UploadFileRequest">
  <wsdl:part name="parameters" element="types:UploadFileRequest"/>
</wsdl:message>
<wsdl:message name="associateResourceResponse">
  <wsdl:part name="parameters" element="types:associateResourceResponse"/>
</wsdl:message>
<wsdl:message name="resumeJobRequest">
  <wsdl:part name="parameters" element="types:resumeJobRequest"/>
</wsdl:message>
<wsdl:message name="rejectJobRequest">
  <wsdl:part name="parameters" element="types:rejectJobRequest"/>
</wsdl:message>
<wsdl:message name="resumeJobResponse">
  <wsdl:part name="parameters" element="types:resumeJobResponse"/>
</wsdl:message>
<wsdl:message name="requestQuoteRequest">
  <wsdl:part name="parameters" element="types:requestQuoteRequest"/>
</wsdl:message>
<wsdl:message name="retrieveFullJobsListRequest">
  <wsdl:part name="parameters" element="types:retrieveFullJobsListRequest"/>
</wsdl:message>
<wsdl:message name="submitJobResponse">
  <wsdl:part name="parameters" element="types:submitJobResponse"/>
</wsdl:message>
<wsdl:message name="retrieveServiceListRequest">
  <wsdl:part name="parameters" element="types:retrieveServiceListRequest"/>
</wsdl:message>
<wsdl:message name="submitJobRequest">
  <wsdl:part name="parameters" element="types:submitJobRequest"/>
</wsdl:message>
<wsdl:message name="retrieveActiveJobsListRequest">
  <wsdl:part name="parameters" element="types:retrieveActiveJobsListRequest"/>
</wsdl:message>
<wsdl:message name="retrieveJobResponse">
  <wsdl:part name="parameters" element="types:retrieveJobResponse"/>
</wsdl:message>
<wsdl:message name="suspendJobRequest">
  <wsdl:part name="parameters" element="types:suspendJobRequest"/>
</wsdl:message>
<wsdl:message name="retrieveJobInformationRequest">
  <wsdl:part name="parameters" element="types:retrieveJobInformationRequest"/>
</wsdl:message>
<wsdl:message name="retrieveFullJobsListResponse">
  <wsdl:part name="parameters" element="types:retrieveFullJobsListResponse"/>
</wsdl:message>
<wsdl:message name="acceptQuoteResponse">
  <wsdl:part name="parameters" element="types:acceptQuoteResponse"/>
</wsdl:message>
<wsdl:message name="cancelJobRequest">
  <wsdl:part name="parameters" element="types:cancelJobRequest"/>
</wsdl:message>
<wsdl:message name="suspendJobResponse">
  <wsdl:part name="parameters" element="types:suspendJobResponse"/>
</wsdl:message>
<wsdl:message name="retrieveJobInformationResponse">
  <wsdl:part name="parameters" element="types:retrieveJobInformationResponse"/>
</wsdl:message>
<wsdl:message name="retrieveQuoteResponse">
  <wsdl:part name="parameters" element="types:retrieveQuoteResponse"/>
</wsdl:message>
<wsdl:message name="retrieveJobRequest">
  <wsdl:part name="parameters" element="types:retrieveJobRequest"/>
</wsdl:message>
<wsdl:message name="cancelJobResponse">
  <wsdl:part name="parameters" element="types:cancelJobResponse"/>
</wsdl:message>
<wsdl:message name="acceptQuoteRequest">
  <wsdl:part name="parameters" element="types:acceptQuoteRequest"/>
</wsdl:message>
<wsdl:message name="associateResourceRequest">
  <wsdl:part name="parameters" element="types:associateResourceRequest"/>
</wsdl:message>
<wsdl:message name="retrieveQuoteRequest">
  <wsdl:part name="parameters" element="types:retrieveQuoteRequest"/>
</wsdl:message>
<wsdl:message name="disassociateResourceResponse">
  <wsdl:part name="parameters" element="types:disassociateResourceResponse"/>
</wsdl:message>
<wsdl:message name="retrieveResourceInformationResponse">
  <wsdl:part name="parameters" element="types:retrieveResourceInformationResponse"/>
</wsdl:message>
<wsdl:message name="retrieveFullResourceListResponse">
  <wsdl:part name="parameters" element="types:retrieveFullResourceListResponse"/>
</wsdl:message>
<wsdl:message name="UploadFileResponse">
  <wsdl:part name="parameters" element="types:UploadFileResponse"/>
</wsdl:message>
<wsdl:message name="retrieveActiveJobsListResponse">
  <wsdl:part name="parameters" element="types:retrieveActiveJobsListResponse"/>
</wsdl:message>

<wsdl:message name="requestQuoteResponse">
  <wsdl:part name="parameters" element="types:requestQuoteResponse"/>
</wsdl:message>

<wsdl:message name="retrieveServiceListResponse">
  <wsdl:part name="parameters" element="types:retrieveServiceListResponse"/>
</wsdl:message>

<wsdl:message name="rejectJobResponse">
  <wsdl:part name="parameters" element="types:rejectJobResponse"/>
</wsdl:message>

<wsdl:portType name="TransWSRequired_PT">
  <wsdl:operation name="submitJob">
    <wsdl:input name="submitJobRequest" message="intf:submitJobRequest"/>
    <wsdl:output name="submitJobResponse" message="intf:submitJobResponse"/>
  </wsdl:operation>

  <wsdl:operation name="retrieveJobInformation">
    <wsdl:input name="retrieveJobInformationRequest" message="intf:retrieveJobInformationRequest"/>
    <wsdl:output name="retrieveJobInformationResponse" message="intf:retrieveJobInformationResponse"/>
  </wsdl:operation>

  <wsdl:operation name="retrieveJob">
    <wsdl:input name="retrieveJobRequest" message="intf:retrieveJobRequest"/>
    <wsdl:output name="retrieveJobResponse" message="intf:retrieveJobResponse"/>
  </wsdl:operation>

  <wsdl:operation name="retrieveActiveJobsList">
    <wsdl:input name="retrieveActiveJobsListRequest" message="intf:retrieveActiveJobsListRequest"/>
    <wsdl:output name="retrieveActiveJobsListResponse" message="intf:retrieveActiveJobsListResponse"/>
  </wsdl:operation>

  <wsdl:operation name="suspendJob">
    <wsdl:input name="suspendJobRequest" message="intf:suspendJobRequest"/>
    <wsdl:output name="suspendJobResponse" message="intf:suspendJobResponse"/>
  </wsdl:operation>

  <wsdl:operation name="resumeJob">
    <wsdl:input name="resumeJobRequest" message="intf:resumeJobRequest"/>
    <wsdl:output name="resumeJobResponse" message="intf:resumeJobResponse"/>
  </wsdl:operation>

  <wsdl:operation name="cancelJob">
    <wsdl:input name="cancelJobRequest" message="intf:cancelJobRequest"/>
    <wsdl:output name="cancelJobResponse" message="intf:cancelJobResponse"/>
  </wsdl:operation>
</wsdl:portType>

<wsdl:portType name="TransWSOptional_PT">
  <wsdl:operation name="retrieveServiceList">
    <wsdl:input name="retrieveServiceListRequest" message="intf:retrieveServiceListRequest"/>
    <wsdl:output name="retrieveServiceListResponse" message="intf:retrieveServiceListResponse"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:operation name="requestQuote">
  <wsdl:input name="requestQuoteRequest" message="intf:requestQuoteRequest"/>
  <wsdl:output name="requestQuoteResponse" message="intf:requestQuoteResponse"/>
</wsdl:operation>

<wsdl:operation name="acceptQuote">
  <wsdl:input name="acceptQuoteRequest" message="intf:acceptQuoteRequest"/>
  <wsdl:output name="acceptQuoteResponse" message="intf:acceptQuoteResponse"/>
</wsdl:operation>

<wsdl:operation name="retrieveQuote">
  <wsdl:input name="retrieveQuoteRequest" message="intf:retrieveQuoteRequest"/>
  <wsdl:output name="retrieveQuoteResponse" message="intf:retrieveQuoteResponse"/>
</wsdl:operation>

<wsdl:operation name="retrieveFullJobsList">
  <wsdl:input name="retrieveFullJobsListRequest" message="intf:retrieveFullJobsListRequest"/>
  <wsdl:output name="retrieveFullJobsListResponse" message="intf:retrieveFullJobsListResponse"/>
</wsdl:operation>

<wsdl:operation name="retrieveFullResourceList">
  <wsdl:input name="retrieveFullResourceListRequest" message="intf:retrieveFullResourceListRequest"/>
  <wsdl:output name="retrieveFullResourceListResponse" message="intf:retrieveFullResourceListResponse"/>
</wsdl:operation>

<wsdl:operation name="rejectJob">
  <wsdl:input name="rejectJobRequest" message="intf:rejectJobRequest"/>
  <wsdl:output name="rejectJobResponse" message="intf:rejectJobResponse"/>
</wsdl:operation>

<wsdl:operation name="associateResource">
  <wsdl:input name="associateResourceRequest" message="intf:associateResourceRequest"/>
  <wsdl:output name="associateResourceResponse" message="intf:associateResourceResponse"/>
</wsdl:operation>

<wsdl:operation name="disassociateResource">
  <wsdl:input name="disassociateResourceRequest" message="intf:disassociateResourceRequest"/>
  <wsdl:output name="disassociateResourceResponse" message="intf:disassociateResourceResponse"/>
</wsdl:operation>

<wsdl:operation name="retrieveResourceInformation">
  <wsdl:input name="retrieveResourceInformationRequest" message="intf:retrieveResourceInformationRequest"/>
  <wsdl:output name="retrieveResourceInformationResponse" message="intf:retrieveResourceInformationResponse"/>
</wsdl:operation>

<wsdl:operation name="UploadFile">
  <wsdl:input name="UploadFileRequest" message="intf:UploadFileRequest"/>
  <wsdl:output name="UploadFileResponse" message="intf:UploadFileResponse"/>
</wsdl:operation>

<wsdl:operation name="retrieveJobListForResource">
Appendix G: Trans WS Types Schema XSD

<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="TransWSTypes.xsd"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:TransWSDomainTypes="TransWSDomainTypes.xsd"
  xmlns:TransWSMimeTypes="TransWSMimeTypes.xsd"
  xmlns:TransWSScope="urn:oasis:names:tc:wstrans:v1:TransWSScope"
  xmlns:TransWSServiceTypes="TransWSServiceTypes.xsd"
  xmlns:TransWSStatus="TransWSStatus.xsd"
  xmlns:iso3166="TransWSCountryCodesISO3166.xsd"
  xmlns:iso639="TransWSLanguageCodesISO639.xsd"
  xmlns:types="TransWSTypes.xsd"
  xmlns:un.ece.unedocs.cl.iso4217="TransWSCurrencyCode.xsd"
  xmlns:xlf="urn:oasis:names:tc:xliff:document:1.1"
  elementFormDefault="qualified"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    schemaLocation="TransWSCountryCodesISO3166.xsd" />
  <xsd:import namespace="urn:oasis:names:tc:wstrans:v1:TransWSDomainTypes"
    schemaLocation="TransWSDomainTypes.xsd" />
  <xsd:import namespace="urn:oasis:names:tc:wstrans:v1:TransWSLanguageCodesISO639"
    schemaLocation="TransWSLanguageCodesISO639.xsd" />
  <xsd:import namespace="urn:oasis:names:tc:wstrans:v1:TransWSMimeTypes"
    schemaLocation="TransWSMimeTypes.xsd" />
  <xsd:import namespace="urn:oasis:names:tc:wstrans:v1:TransWSScope"
    schemaLocation="TransWSScope.xsd" />
  <xsd:import namespace="urn:oasis:names:tc:wstrans:v1:TransWSServiceTypes"
    schemaLocation="TransWSServiceTypes.xsd" />
  <xsd:import namespace="urn:oasis:names:tc:wstrans:v1:TransWSStatus"
    schemaLocation="TransWSStatus.xsd" />
  <xsd:import namespace="TransWSCurrencyCode.xsd"
    schemaLocation="TransWSCurrencyCode.xsd" />
  <xsd:import namespace="urn:oasis:names:tc:xliff:document:1.1"
    schemaLocation="xliff-1.1.xsd" />
</schema>
<xsd:sequence />
</xsd:complexType>
</xsd:element>
<xsd:element name="retrieveServiceListResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:languagePairs" />
      <xsd:element ref="types:services" />
      <xsd:element ref="types:mimeTypes" />
      <xsd:element ref="types:domains" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="locale">
  <xsd:complexType>
    <xsd:attribute name="country" type="iso3166:CodeContentType_3166" use="required" />
    <xsd:attribute name="language" type="iso639:CodeContentType_639" use="required" />
    <xsd:attribute name="variant" type="xsd:string" use="optional" />
  </xsd:complexType>
</xsd:element>
<xsd:element name="source">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:locale" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="target">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:locale" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="languagePair">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:source" />
      <xsd:element ref="types:target" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="languagePairs">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="1" ref="types:languagePair" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="service" type="TransWSServiceTypes:CodeContentType_service" />
<xsd:element name="services">
  <xsd:complexType>
<xsd:sequence>
  <xsd:element maxOccurs="unbounded" minOccurs="1" ref="types:service" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="mimeType" type="TransWSMimeTypes:CodeContentType_mime" />
<xsd:element name="mimeTypes">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="1" ref="types:mimeType" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="domain" type="TransWSDomainTypes:CodeContentType_domain" />
<xsd:element name="domains">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="1" ref="types:domain" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!--
  -- requestQuote
  -->
<xsd:element name="requestQuoteRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket" />
      <xsd:element ref="types:count-group" />
      <xsd:element ref="types:service" />
      <xsd:element ref="types:domain" />
      <xsd:element ref="types:languagePair" />
      <xsd:element ref="types:requiredBy" />
      <xsd:element ref="types:info" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="requestQuoteResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket" />
      <xsd:element ref="types:available" />
      <xsd:element ref="types:fileUri"/>
    </xsd:sequence>
    <!--
      <xsd:element ref="types:quotation"/>
      <xsd:element ref="types:expires"/>
      <xsd:element ref="types:info"/>
    -->
  </xsd:complexType>
</xsd:element>
<xsd:element name="retrieveQuoteRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="retrieveQuoteResponse">
  <xsd:complexType>
    <xsd:sequence>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="acceptQuoteRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket" />  
      <xsd:element ref="types:purchaseOrder" />
      <xsd:element ref="types:purchaseOrderUri" use="optional" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

</xsd:element>

<xsd:element name="acceptQuoteResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket" />
      <xsd:element ref="types:acknowledgement" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="purchaseOrder">
  <xsd:complexType>
    <xsd:attribute name="id" type="string" use="required" />
  </xsd:complexType>
</xsd:element>

<xsd:element name="purchaseOrderUri">
  <xsd:complexType>
    <xsd:attribute name="uri" type="anyURI" use="optional" />
  </xsd:complexType>
</xsd:element>

<!--
  -- submitJob
-->

<xsd:element name="submitJobRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket" />
      <xsd:element ref="types:count-group" />
      <xsd:element ref="types:service" />
      <xsd:element ref="types:domain" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="event">
  <xsd:complexType>
    <xsd:attribute name="timestamp" type="xsd:dateTime" use="required" />
    <xsd:attribute name="stage" type="xsd:string" use="required" />
    <xsd:attribute name="status" type="TransWSStatus:CodeContentType_status" use="required" />
    <xsd:attribute name="date" type="dateTime" use="optional" />
  </xsd:complexType>
</xsd:element>

<!--
  - - retrieveJob
-->
<xsd:element name="retrieveJobRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="retrieveJobResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket" />
      <xsd:element ref="types:resource" />
      <xsd:element ref="types:rejectionDeadline" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="rejectionDeadline">
  <xsd:complexType>
    <xsd:attribute name="date" type="dateTime" use="optional" />
  </xsd:complexType>
</xsd:element>

<!--
  - - rejectJob
-->
<xsd:element name="rejectJobRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket" />
      <xsd:element ref="types:reason" />
      <xsd:element ref="types:info" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="rejectJobResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket" />
      <xsd:element ref="types:acknowledgement" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="reason" type="xsd:string" />

<!--
   - - retrieveActiveJobsList
-->
<xsd:element name="retrieveActiveJobsListRequest">
  <xsd:complexType>
    <xsd:sequence />
  </xsd:complexType>
</xsd:element>
<xsd:element name="retrieveActiveJobsListResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" ref="types:ticket" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!--
   - - retrieveFullJobsList
-->
<xsd:element name="retrieveFullJobsListRequest">
  <xsd:complexType>
    <xsd:sequence />
  </xsd:complexType>
</xsd:element>
<xsd:element name="retrieveFullJobsListResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" ref="types:ticket" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!--
   - - cancelJob
-->
<xsd:element name="cancelJobRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="cancelJobResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket" />  
      <xsd:element ref="types:succeeded" />  
      <xsd:element ref="types:info" />  
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="succeeded" type="xsd:boolean" />

<!--
-- suspendJob
-->
<xsd:element name="suspendJobRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket" />  
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="suspendJobResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket" />  
      <xsd:element ref="types:succeeded" />  
      <xsd:element ref="types:info" />  
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!--
-- resumeJob
-->
<xsd:element name="resumeJobRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket" />  
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="resumeJobResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket" />  
      <xsd:element ref="types:succeeded" />  
      <xsd:element ref="types:info" />  
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!--
-- associateResource
-->

<xsd:element name="associateResourceRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket" />
      <xsd:element ref="types:resource" />
      <xsd:element ref="types:purpose" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="purpose">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:attribute name="usage" type="xsd:string" use="required"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="associateResourceResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket" />
      <xsd:element ref="types:succeeded" />
      <xsd:element ref="types:resource" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!--
  - - disassociateResource
-->
<xsd:element name="disassociateResourceRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket" />
      <xsd:element ref="types:resource" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="disassociateResourceResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket" />
      <xsd:element ref="types:succeeded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!--
  - - retrieveJobListForResource
-->
<xsd:element name="retrieveJobListForResourceRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:resource" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="retrieveJobListForResourceResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" ref="types:ticket"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!--
- - retrieveResourceInformation
-->}

<xsd:element name="retrieveResourceInformationRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:resource"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="retrieveResourceInformationResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="types:ticket"/>
      <xsd:element ref="types:purpose"/>
      <xsd:element ref="types:resource"/>
      <xsd:element ref="types:resource"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!--
- - retrieveFullResourceList
-->}

<xsd:element name="retrieveFullResourceListRequest">
  <xsd:complexType>
    <xsd:sequence/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="retrieveFullResourceListResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" ref="types:resource"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!--
- - UploadFile
-->}

<xsd:element name="UploadFileRequest"
Appendic H: References

Normative

[IANA Charsets] IANA Names for Character Sets. IANA (Internet Assigned Numbers Authority), Aug 2001
[WSDL] Web Services Description Language. Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001

Non-Normative

Appendix I: Glossary

**OASIS**
Organization for the Advancement of Structured Information Standards.

**SOAP**

**XLIFF**
OASIS Standard for XML Localization Interchange File Format.

**XML**
eXtensible Markup Language.

**WSBPEL**
OASIS Web Services Business Process Execution Language.

**WSDL**
Web Services Description Language.