

Oracle Taleo Enterprise Integration

Bulk API Reference Guide

Feature Pack 14B

Part Number: E58305-01

November 2014

Bulk API Reference Guide

Part Number: E58305-01

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Saurabh Agarwal

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Bulk API Overview	
Overview.....	2
Quick Start Guide	
Quick Start Guide.....	5
Developer's Workbook	
Developer's Workbook.....	12
Data Import and Export	
Data Import and Export.....	15
Import.....	17
Export.....	21
Exceptions and Error Handling	
Exceptions and Error Handling.....	26
Bulk API Security	
Bulk API Security.....	29
MTOM Guide	
Using MTOM.....	31
Troubleshooting Guide	

Troubleshooting Overview.....	42
Request/Response XMLs in TCC.....	42
Downloading Error Logs.....	43

Appendix

Volume Limits.....	46
List of Possible Errors.....	47
Taleo Connect Server.....	54

Bulk API Overview

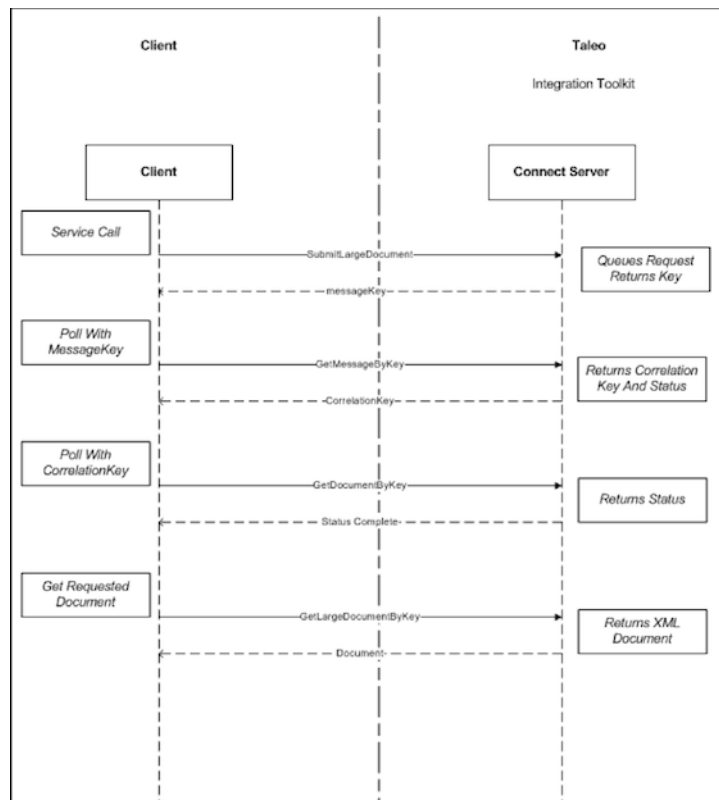
• Overview.....	2
-----------------	---

Overview

Bulk API is a set of web service operations for bulk import and export of data to and from Taleo. These operations enable the transfer of large amounts of data asynchronously without impacting product performance.

Process Flow

The following figure shows a sequence of web service calls for transfer of data between the client and Taleo system. The client initiates a web service call on Taleo Integration Toolkit. Taleo Connect Server queues the request and returns a key. The client polls the Taleo Connect Server using this key until the server returns a correlation key. The client then polls the server with the correlation key until the status of the response document is complete. The client then retrieves the response document using the correlation key.



Technologies Tested

Bulk API has been implemented as a SOAP web service and tested on the following technology stacks. Most of the code sample in this documentation is available using CXF (WSDL2JAVA) with JAXB bindings using Eclipse IDE.

Table 1: Technologies Tested

Language	Stub Generation	WS Stack	Binding	Supported	Comment
SOAP UI	N/A	N/A	N/A	Yes	
Java	CXF WSDL to Java	CXF	JAXB	Yes	

Language	Stub Generation	WS Stack	Binding	Supported	Comment
Java	CXF WSDL to Java	CXF	XMLBean	Limited (no attachment support)	XML Bean does not support MTOM attachment
.NET C#		WS		No	
.NET C#	WseWsd13.exe	WSE		No	Required – works at C programming level
.NET C#	Svcutil.exe	WCF		Yes	
Others				No	

Standards Compliance

The API conforms to the following standards:

Standard Name	Website
Simple Object Access Protocol (SOAP) 1.1	http://www.w3.org/TR/2000/NOTE-SOAP-20000508
Web Service Description Language (WSDL) 1.1	http://www.w3.org/TR/2001/NOTE-wsdl-20010315
Hypertext Transfer Protocol (HTTP) 1.1	http://www.w3.org/Protocols/rfc2616/rfc2616.html
Web Services Security 1.1	http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf

Quick Start Guide

• Quick Start Guide.....	5
--------------------------	---

Quick Start Guide

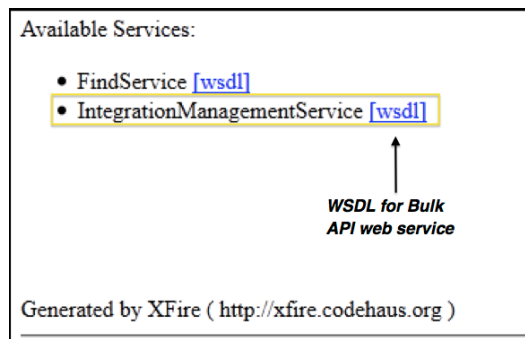
WSDL

The WSDL for Bulk API web service can be obtained from the zone information by using the following taxonomy – [protocol]//[zone]/[product-code]/soap:

- Replace [zone] with your zone information.
- Replace [product-code] with the required product. The products are listed in the table below.

Type the URL (which will be similar to <https://myzone.taleo.net/smartorg/soap>) in a browser. Enter the account credentials. Select the WSDL for Bulk API web service by clicking WSDL next to IntegrationManagementService.

<https://yourzone.taleo.net/product/soap>



Product	
Name	Code
SmartOrg	smartorg
Recruiting	enterprise
Onboarding	transitions
Performance	orion



Request System Administrator for a user account with “System Integrator” role.

Web Service Operations

Name	Description
submitDocument()	This operation has been deprecated.
submitLargeDocument()	Most frequently used operation. When submitting XML documents greater than 50kb, the document must be attached using MTOM. See MTOM Request Sample in the MTOM chapter.

Name	Description
	If the XML payload is smaller than 50kb, it can be included in the SOAP envelope as Base 64 encoded content. Sending an XML document larger than 50kb as base 64 content will result in error.
getMessageByIdentifier()	This operation has been deprecated.
getMessageByKey()	The operation is used for polling both the request and the response documents. It can take the messagekey as the parameter and return the status of the request or it can take the correlationkey and return the status of the response.
getDocumentByKey()	This operation has been deprecated.
getLargeDocumentByKey()	The operation returns the information requested as an attachment. It uses MTOM for message attachment and streaming.

Operation Calls in Detail

Table 2: submitLargeDocument()

Operation Signature
ns2011_05:IntegrationMessage : submitLargeDocument(wsa:MessageID, wsa:ReplyTo, wsa:Action, tns:Document)
Sample Request
<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:add="http://www.w3.org/2005/03/addressing" xmlns:man="http://www.taleo.com/ws/integration/toolkit/2011/05/management" xmlns:quer="http://www.taleo.com/ws/integration/query"> <soapenv:Header> <!--Input parameter--> <!--Indicates how Taleo Connect Server processes the included document--> <add:Action>http://www.taleo.com/ws/integration/toolkit/2005/07/action/export </add:Action> <!--Input parameter--> <!--Indicates how Taleo Connect Server returns the corresponding response message--> <add:ReplyTo> <add:Address>http://www.taleo.com/ws/integration/toolkit/2005/07/addressing/queue </add:Address> <add:ReferenceParameters> </add:ReferenceParameters> </add:ReplyTo> <!--Input parameter--> <!--Client provided value for associating the call with the response--> <add:MessageID>Message_ID_3</add:MessageID> </soapenv:Header> <soapenv:Body> <man:submitLargeDocument></pre>

```

        <!--Input parameter-->
        <!--Reference to MTOM attachment or can contains a base64 encoded
Document-->
        <man:Document>cid:submitLargeDocFile.xml</man:Document>
    </man:submitLargeDocument>
</soapenv:Body>
</soapenv:Envelope>

```

Sample Attachment

```

<Document xmlns="http://www.taleo.com/ws/integration/toolkit/2011/05">
  <Attributes>
    <Attribute name="version">http://www.taleo.com/ws/tee800/2009/01</
Attribute>
    <Attribute name="mode">T-XML</Attribute>
  </Attributes>
  <Content>
    <ExportQuery xmlns="http://www.taleo.com/ws/integration/toolkit/2005/07/
action/export">
      <quer:query projectedClass="Application" alias="NewHire"
preventDuplicates="false"
        xmlns:quer="http://itk.taleo.com/ws/query">
        <quer:subQueries/>
        <quer:projections>
          <quer:projection alias="Username">
            <quer:field path="Candidate,Credentials,Username"/>
          </quer:projection>
          <quer:projection alias="Requisition_Number">
            <quer:field path="Requisition,ContestNumber"/>
          </quer:projection>
          :
          :
          :
        <quer:groupings/>
        <quer:joinings/>
      </quer:query>
    </ExportQuery>
  </Content>
</Document>

```

Sample Response

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <ns1:submitLargeDocumentResponse xmlns:ns1="http://www.taleo.com/ws/
integration/
      toolkit/2011/05/management">
      <IntegrationMessage xmlns="http://www.taleo.com/ws/integration/
toolkit/2011/05">
        <!--Output parameter-->
        <!--A globally unique identifier generated by the Taleo Connect
Server-->
          <MessageKey>1828611696</MessageKey>
        </ns1:submitLargeDocumentResponse>
      </soap:Body>

```

```
</soap:Envelope>
```

Table 3: getMessageByKey()

Operation Signature
ns2011_05:IntegrationMessage : getMessageByKey(tns:messageKey)
Sample Request
<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:man="http://www.taleo.com/ws/integration/toolkit/2011/05/management"> <soapenv:Header/> <soapenv:Body> <man:getMessageByKey> <!--Input parameter--> <!--Id returned in response to submitLargeDocument()--> <man:messageKey>1828612469</man:messageKey> </man:getMessageByKey> </soapenv:Body> </soapenv:Envelope></pre>
Sample Response
<pre><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <soap:Body> <!--Output parameter--> <ns1:getMessageByKeyResponse xmlns:ns1="http://www.taleo.com/ws/integration/ toolkit/2011/05/management"> <IntegrationMessage xmlns="http://www.taleo.com/ws/integration/ toolkit/2011/05"> <!--ID returned by the SubmitLargeDocument operation response.--> <MessageKey>1828612469</MessageKey> <!--Unique key used internally within the Taleo product instance--> <CorrelationKey>1828612448</CorrelationKey> <!--Globally unique value generated by Taleo Connect Server--> <MessageID>bea441d0-9c87-11e2-af2f-f3cf7ee15a07</MessageID> <ConversationID/> <State>5</State> <!--processing state within the Taleo Connect server--> <Target>4</Target> <Format>2</Format> <ServiceName>null</ServiceName> <MethodName>null</MethodName> <ServiceID>http://www.taleo.com/ws/integration/ toolkit/2005/07/action/ export</ServiceID> <ApplicationCode>enterprise</ApplicationCode> <CreationDate>2013-04-03T13:56:01.000-04:00</CreationDate></pre>

```

        <LastModificationDate>2013-04-03T13:56:03.000-04:00</
LastModificationDate>
        <OwnerUserName>support</OwnerUserName>
        <Document>
            <RecordCount>0</RecordCount>
            <RecordIndex>0</RecordIndex>
            <Attributes>
                <Attribute>
                    <Name>duration</Name>
                    <Value>0:00:00.006</Value>
                </Attribute>
            </Attributes>
        </Document>
        <HistoryItems>
            <HistoryItem>
                <Sequence>1</Sequence>
                <!--Order of change in states-->
                <InitialState>1</InitialState>
                <!--Initial state in this step-->
                <FinalState>1</FinalState>
                <!--Final state in this step-->
                <UserName>support</UserName>
                <CreationDate>2013-04-03T13:56:01.000-04:00</
CreationDate>
                <!--Timestamp of execution of this step-->
            </HistoryItem>
            <HistoryItem>
                <Sequence>2</Sequence>
                <InitialState>1</InitialState>
                <FinalState>2</FinalState>
                <UserName>support</UserName>
                <CreationDate>2013-04-03T13:56:01.000-04:00</
CreationDate>
            </HistoryItem>
            <HistoryItem>
                :
                :
                :
            </HistoryItem>
        </HistoryItems>
    </IntegrationMessage>
</nsl:getMessageByKeyResponse>
</soap:Body>
</soap:Envelope>

```

Table 4: getLargeDocumentByKey()

Operation Signature
tns:Document : getLargeDocumentByKey (tns:messageKey)
Sample Request
<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/ envelope/" xmlns:man="http://www.taleo.com/ws/integration/toolkit/2011/05/ management"> <soapenv:Header/> </pre>

```

<soapenv:Body>
  <man:getLargeDocumentByKey>
    <!--Input parameter, this is the Correlation ID returned in
getMessageByKey
      operation call-->
    <man:messageKey>1828611678</man:messageKey>
  </man:getLargeDocumentByKey>
</soapenv:Body>
</soapenv:Envelope>

```

Sample Response

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <!--Output-->
    <ns1:getLargeDocumentByKeyResponse
      xmlns:ns1="http://www.taleo.com/ws/integration/
toolkit/2011/05/management">
      <ns1:Document ns2:contentType="text/xml; charset=UTF-8"
        xmlns:ns2="http://www.w3.org/2004/11/xmlmime">
        <!--Reference to MTOM attachment-->
        <Include href="cid:136512867531824-150371349@http://
www.w3.org/2001/XMLSchema"
          xmlns="http://www.w3.org/2004/08/xop/include"/>
        </ns1:Document>
      </ns1:getLargeDocumentByKeyResponse>
    </soap:Body>
  </soap:Envelope>

```



- Only the Output/Input parameters and key elements are presented in above XML snippets
- For details on the elements and their use, see [Message Processing](#) in the Appendix
- A SoapUI project is available for running the above operations

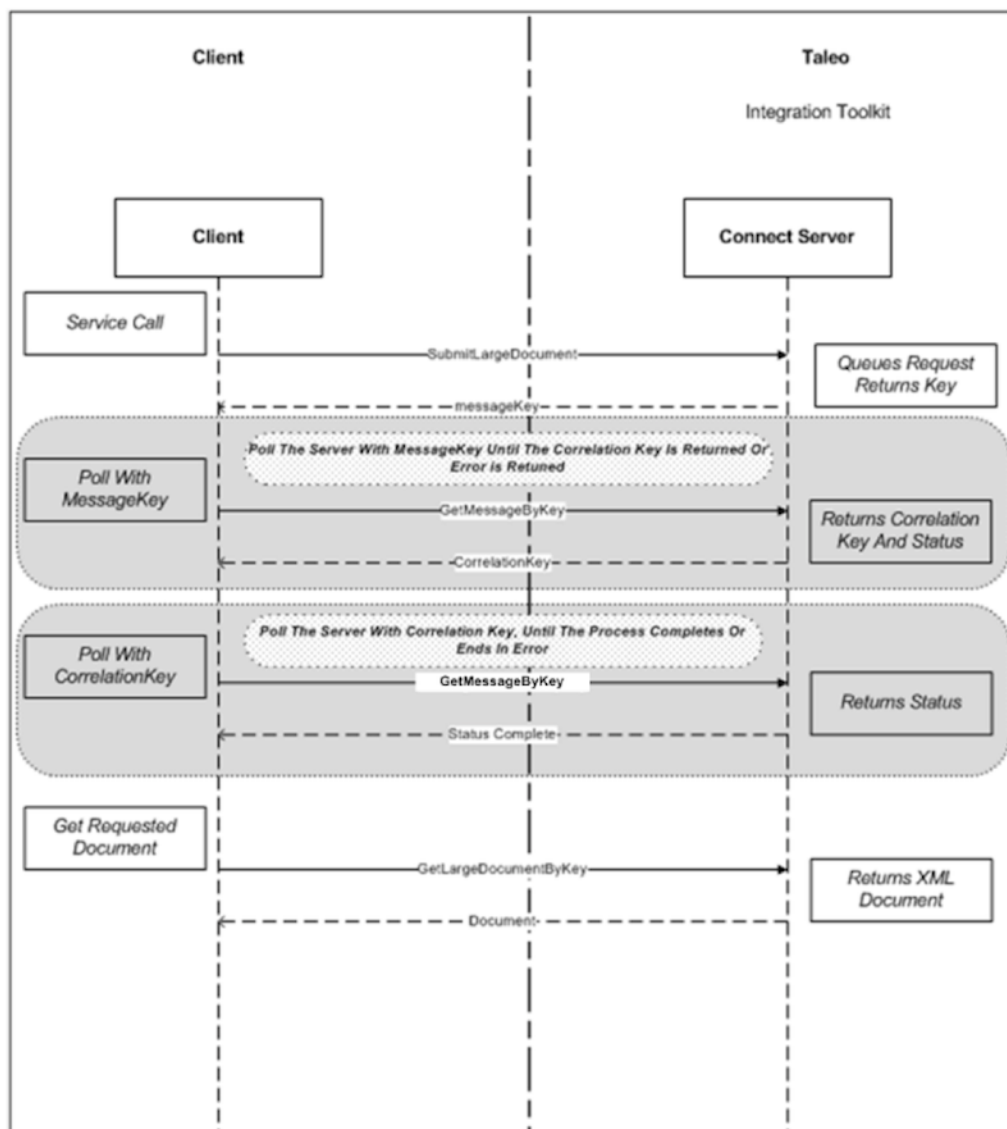
Developer's Workbook

• Developer's Workbook.....	12
-----------------------------	----

Developer's Workbook

Using Bulk API

The previous sections describe the Bulk API and the operation calls in general. This section focuses on the sequence of Bulk API operation calls and usage guidelines. The figure below illustrates the flow of Bulk API operation calls for a standard implementation.



1. Submitting the request: As a first step the client submits a request by calling submitLargeDocument operation. Taleo Connect Server adds the request in the queue and returns a MessageKey in response. It is recommended that client provides a unique MessageID value (wsa:MessageID) in the SOAP header. This value can greatly assist in troubleshooting during the implementation phase. The request and response corresponding to the MessageID can be easily tracked in TCC Console as well in system logs when troubleshooting errors.

2. Polling to get the status of request: Client polls the server to get the status of the request by calling the operation `GetMessageByKey`. This operation should be called using the `MessageKey` returned by the server in the first step. The client should continue polling until the server returns a `CorrelationKey`. Once the `CorrelationKey` has been returned, client must continue polling the server by calling the operation `GetMessageByKey` and passing the `CorrelationKey` until the process completes or ends in error. The polling interval must be adjusted to avoid unnecessary polling on the zone. Each integration request must be analyzed based on size of data transfer and complexity to set the polling interval optimally. The possible states are `COMPLETED`, `INTERRUPTED`, `INERROR`, `SUSPENDED`.
3. Retrieving the requested document: Finally, the client calls the operation `getLargeDocumentByKey` passing the `CorrelationKey` as the parameter. Taleo Connect Server returns the requested document.

The preceding sections describe Bulk API architecture, operations and implementation requirements. The following sections in this documentation discuss the details of the XML payload that is transferred via these operations.

Web Service in Eclipse IDE

Sample java code for calling the Bulk API is included with this documentation. The project includes detailed documentation on environment set-up and steps for executing the code.

System Requirements:

- Eclipse IDE
- Apache Ant
- JRE 1.6 or higher
- Apache CXF 2.6.7

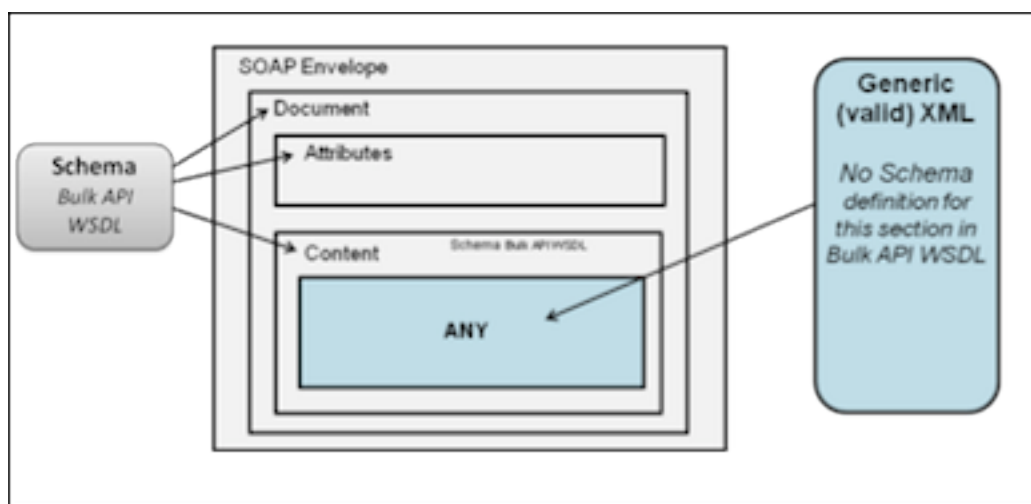
Data Import and Export

- Data Import and Export..... 15

Data Import and Export

Structure of Data Transferred by Bulk API Call

The section of XML within <Document> tags contains the data and the information on the desired operation on this data. While Document/Attributes and Document/Content element are included in the Bulk API WSDL and Schema, there are no details available for this section in the WSDL. This section of the documentation describes the structure of data and the operations that can be executed on this data. The figure below illustrates the structure of the SOAP.



Structure of a Document Tag

For details on all the attributes and elements and their possible values, please see [Document Attributes](#) and [Document Elements](#) in the Appendix.

See [Import](#) and [Export](#) sections below for the *necessary* changes needed to XML from TCC before it can be used with Bulk API calls.

The following xml snippet illustrates the Document element and its sub-elements.

```
<Document xmlns="http://www.taleo.com/ws/integration/toolkit/2011/05">
  <Attributes>
    <Attribute name="locale">en</Attribute>
    <Attribute name="largegraph">true</Attribute>
    <Attribute name="mode">CSV-ENTITY</Attribute>
    <Attribute name="version">http://www.taleo.com/ws/so800/2009/01</
Attribute>
  </Attributes>
  <Content>
    <ExportQuery xmlns="http://www.taleo.com/ws/so800/2009/01">
      <quer:query xmlns="http://www.taleo.com/ws/integration/
 toolkit/2005/07"
 xmlns:m="http://www.taleo.com/ws/integration/
 toolkit/2005/07"
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:tn="http://www.taleo.com/ws/integration/tn/2005/07"
 xmlns:quer="http://www.taleo.com/ws/integration/query"
 xmlns:wsa="http://www.w3.org/2005/03/addressing">
```

```

        alias="Query466FE670-708D-11E2-9810-F47AEB6ADAA3"
        preventDuplicates="false" projectedClass="User">
<quer:subQueries/>
<quer:projections>
  <quer:projection>
    <quer:field path="FirstName"/>
  </quer:projection>
  <quer:projection>
    <quer:field path="LastName"/>
  </quer:projection>
  <quer:projection>
    <quer:field path="Department,Code"/>
  </quer:projection>
  <quer:projection>
    <quer:field path="Department,Name"/>
  </quer:projection>
</quer:projections>
<quer:projectionFilterings/>
<quer:filterings/>
<quer:sortings/>
<quer:sortingFilterings/>
<quer:groupings/>
<quer:joinings/>
  </quer:query>
</ExportQuery>
</Content>
</Document>

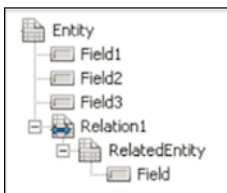
```

The request for transfer of data is contained in the query within the content tags. This content can be easily generated using the integration tool Taleo Connect Client (TCC). The TCC User Guide is available on OTN at <http://www.oracle.com/technetwork/documentation/taleoenterprise-1648220.html#platform>. The details on creating import and export scripts are available in Getting Started section of the TCC User Guide. This guide explains in detail the entities and their fields. The _sq.xml file generated by TCC can be used as the content for export or import request.

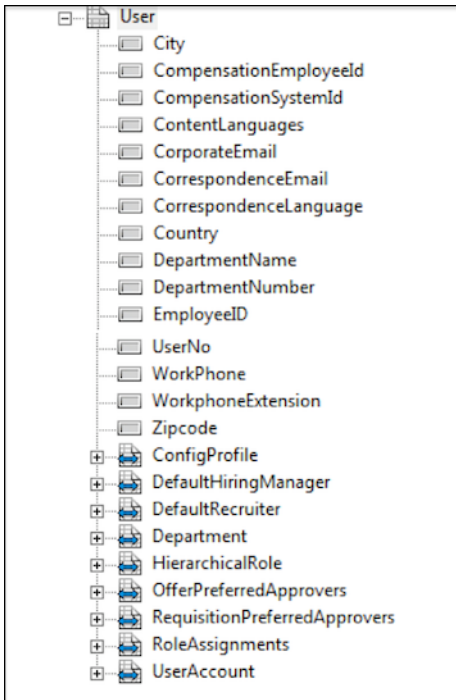
Data Structure

The data in Taleo is architected in an entity relationship model. It is based on the following three concepts:

- Entity: A basic object unit that represents a general business concept
- Field: A characteristic or attribute of an entity
- Relation: A link between two entities



Structure of Entity



A Sample User Entity

Import

The import processes enables "create" and/or "update" actions on the business entities within the Taleo zone. This section describes the process of constructing an import instruction followed by import document and finally wrapping the document in a T-SOAP message.

Import Instructions

Creating a basic import instruction is quite straightforward. The import process enables *create*, *update*, *delete*, *move*, or any specialized action on the business process. For example, to create a new user in the zone with the name John Smith and personal email jsmith@acme.com, the instruction would be:

```
<User>
  <PersonalEmail>jsmith@acme.com</PersonalEmail>
  <FirstName>John</FirstName>
  <LastName>Smith</LastName>
</User>
```

An update instruction can be created as easily. In this case, search attribute must be specified as criteria to find the existing entity. The following instruction searches the user based on the e-mail address and updates the LastName field (leaving the FirstName field value as is):

```
<User>
  <PersonalEmail searchType="search" searchValue="jsmith@acme.com"/>
  <LastName>Smith</LastName>
</User>
```

The default behavior of an import instruction is to merge the information provided. That is, if the entity is found, then an update is made. If the entity is not found, then it is created with the information provided. In certain cases, the value used to determine the entity can also be updated. The syntax of such instruction is presented below:

```
<User>
  <PersonalEmail searchType="searchAndValue"
    searchValue="jsmith@acme.com">jbrown@acme.com"</PersonalEmail>
</User>
```

Field types. There are five basic field types used in Taleo business models: text, numeric, date, Boolean, and multilingual. Dates must be provided in the ISO-8601 format, for example:

```
<DateOfEntry>2006-06-01T14:15:00-04:00</DateOfEntry>
```

Multilingual values are provided individually by locale, for example:

```
<Location>
  <Code>USA</Code>
  <Description>
    <value locale='en'>United States of America</value>
    <value locale='fr'>États-Unis d'Amérique </value>
    <value locale='es'>Estados Unidos de América </value>
  </Description>
</Location>
```

Entity Relation

When importing entities, most relations are lookups and the related entity is only linked to the main entity. This is the case of the Department relation of the User entity. To determine the related entity, the same search attribute is used as above, but in a different context. For example, to specify John Smith in department 10001, we would write:

```
<User>
  <FirstName>John</FirstName>
  <LastName>Smith</LastName>
  <Department>
    <Department>
      <Code searchType="search" searchValue="10001"/>
    </Department>
  </Department>
</User>
```

In the above example, the location is searched on the code (USA in this case) and associated with the candidate John Smith being created.

Custom Fields (user defined fields (UDF))

Custom fields can be handled similarly.

```
<User>
  <PersonalEmail searchType="search" searchValue=jsmith@acme.com/>
  <UDFs>
    <UDF name="DriverLicenseNumber">JS-552-87610</UDF>
```

```
</UDFs>
</User>
```

Custom fields can be multilingual and can be managed the same way as the standard fields.

```
<User>
  <PersonalEmail searchType="search" searchValue=jsmith@acme.com/>
  <UDFs>
    <UDF name="CertificationInformation">
      <value locale="en">Chartered Accountant, level 2</value>
    </UDF>
  </UDFs>
</User>
```

The custom fields can be imported by searching within the custom fields. However, it is not possible to search for any entity based on a custom field.

```
<User>
  <CorrespondenceEmail>tester@invalidemail.com</CorrespondenceEmail>
  <EmployeeID>10001</EmployeeID>
  <FirstName>testfirst</FirstName>
  <LastName>testlast</LastName>
  <ContentLanguages>en</ContentLanguages>
  <CorrespondenceLanguage>en</CorrespondenceLanguage>
  <Title>
    <value locale="en">Test-title</value>
  </Title>
  <UserAccount>
    <UserAccount>
      <Loginname>Tester</Loginname>
      <ForceChangePassword>yes</ForceChangePassword>
    </UserAccount>
  </UserAccount>
  <UDFs>
    <UDF name="PIn_20_2d_20Career_20Section_20Only">
      <UDSElement>
        <Code searchType="search" searchTarget="." searchValue="1"/>
      </UDSElement>
    </UDF>
  </UDFs>
</User>
```

Import Document

The following is an example of Import Document. Here the mandatory attribute is **version** of the model used, in our case <http://www.taleo.com/ws/tee800/2009/01>. The document also has the namespace specified.

```
<Document xmlns="http://www.taleo.com/ws/integration/toolkit/2011/05">
  <Attributes>
    <Attribute name="version">http://www.taleo.com/ws/so800/2009/01</
Attribute>
  </Attributes>
  <Content>
    <ImportEntities xmlns="http://www.taleo.com/ws/so800/2009/01">
      <User-create xmlns="http://www.taleo.com/ws/so800/2009/01"
xmlns:m="http://www.taleo.com/ws/integration/
toolkit/2005/07"
```

```

xmlns:SOAP-ENV = "http://schemas.xmlsoap.org/soap/
envelope/"
xmlns:tn="http://www.taleo.com/ws/integration/
tn/2005/07"
xmlns:wsa="http://www.w3.org/2005/03/addressing">
  <User>
    <CorrespondenceEmail>tester@invalidemail.com</
CorrespondenceEmail>
    <EmployeeID>10001</EmployeeID>
    <FirstName>testfirst</FirstName>
    <LastName>testlast</LastName>
    <ContentLanguages>en</ContentLanguages>
    <CorrespondenceLanguage>en</CorrespondenceLanguage>
    <Title>
      <value locale="en">Test-title</value>
    </Title>
    <UserAccount>
      <UserAccount>
        <Loginname>Tester</Loginname>
        <ForceChangePassword>yes</ForceChangePassword>
      </UserAccount>
    </UserAccount>
  </User>
  <Password>test1234</Password>
  <Null/>
</User-create>
</ImportEntities>
</Content>
</Document>

```



It is mandatory to specify the version explicitly as an attribute.

Import Message

The import message wraps the document within a standard SOAP envelope and adds some header elements required for the T-SOAP specifications. The most important element is the **action** element that indicates Taleo Connect Server to interpret the integration instructions as import.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/
envelope/"
  xmlns:add="http://www.w3.org/2005/03/addressing"
  xmlns:man="http://www.taleo.com/ws/integration/toolkit/2011/05/
management"
  xmlns:ns="http://www.taleo.com/ws/integration/toolkit/2011/05"
  xmlns:quer="http://www.taleo.com/ws/integration/query">
  <soapenv:Header>
    <!--The Action must be specified correctly -->
    <add:Action>http://www.taleo.com/ws/integration/toolkit/2005/07/action/
import</add:Action>
    <add:ReplyTo>
      <add:Address>http://www.taleo.com/ws/integration/
 toolkit/2005/07/
        addressing/queue</add:Address>
      <add:ReferenceParameters></add:ReferenceParameters>
    </add:ReplyTo>
    <add:MessageID>Message_ID_101</add:MessageID>
  </soapenv:Header>
  <soapenv:Body>
    <man:submitDocument>

```



```

        <Document>
        <!--The document element was discussed in the sample above -->
        </ns:Document>
    </man:submitDocument>
</soapenv:Body>
</soapenv:Envelope>

```



It is mandatory to specify the action element correctly.

Export

The export feature processes a single integration instruction to extract data from the zone. This section illustrates creation of export instruction, document, and message.

Export Instructions

The export instruction is built using SQ-XML only. However, the output can be generated in four formats XML, T-XML, CSV, and CSV-Entity. The export instruction specifies the entity and the fields of the entity being extracted. The following sample illustrates the same:

```

<quer:query alias="Query41312D38-68E8-11E2-A47A-A9EE1DAC0539 "
  preventDuplicates="false"
  projectedClass="User" xmlns="http://www.taleo.com/ws/integration/
  toolkit/2005/07"
  xmlns:m="http://www.taleo.com/ws/integration/toolkit/2005/07"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:tn="http://www.taleo.com/ws/integration/tn/2005/07"
  xmlns:quer="http://www.taleo.com/ws/integration/query"
  xmlns:wsa="http://www.w3.org/2005/03/addressing">
  <quer:subQueries/>
  <quer:projections>
    <quer:projection>
      <quer:field path="FirstName"/>
    </quer:projection>
    <quer:projection>
      <quer:field path="LastName"/>
    </quer:projection>
    <quer:projection>
      <quer:field path="EmployeeID"/>
    </quer:projection>
    <quer:projection>
      <quer:field path="DepartmentName"/>
    </quer:projection>
    <quer:projection>
      <quer:field path="CorrespondenceEmail"/>
    </quer:projection>
    <quer:projection>
      <quer:field path="StreetAddress"/>
    </quer:projection>
  </quer:projections>
  <quer:projectionFilterings/>
  <quer:filterings>
    <quer:filtering>
      <quer:equal>
        <quer:field path="LastName"/>
        <quer:string>Cannon</quer:string>
      </quer:equal>
    </quer:filtering>
  </quer:filterings>
</quer:query>

```

```

    </quer:filtering>
  </quer:filterings>
</quer:sortings/>
</quer:sortingFilterings/>
</quer:groupings/>
</quer:joinings/>
</quer:query>

```

Export Document

The following sample describes the export document. The following attributes are required: locale, largegraph, mode, version.

```

<Document xmlns="http://www.taleo.com/ws/integration/toolkit/2011/05">
  <!-- The attributes defined here are mandatory; incorrect values will
        result in document processing error. -->
  <Attributes>
    <Attribute name="locale">en</Attribute>
    <Attribute name="largegraph">>true</Attribute>
    <Attribute name="mode">T-XML</Attribute>
    <Attribute name="version">http://www.taleo.com/ws/so800/2009/01</
Attribute>
  </Attributes>
  <Content>
    <ExportQuery xmlns="http://www.taleo.com/ws/so800/2009/01">
      <quer:query alias="Query752BAE88-55BD-11E2-8B1D-CDAFDC756D0C"
        preventDuplicates="false" projectedClass="User"
        xmlns="http://www.taleo.com/ws/integration/
toolkit/2005/07"
        xmlns:m="http://www.taleo.com/ws/integration/
toolkit/2005/07"
        xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/
envelope/"
        xmlns:tn="http://www.taleo.com/ws/integration/
tn/2005/07"
        xmlns:quer="http://www.taleo.com/ws/integration/query"
        xmlns:wsa="http://www.w3.org/2005/03/addressing">
        <quer:subQueries/>
        <quer:projections>
          <quer:projection>
            <quer:field path="CorrespondenceEmail"/>
            <quer:filterings>
              <quer:filtering>
                <quer:equal>
                  <quer:field path="CorrespondenceEmail"/
>
                  <quer:string>test-
clientname@testorg.com</quer:string>
                </quer:equal>
                :
                :
              </quer:filtering>
            </quer:filterings>
          </quer:projection>
        </quer:projections>
      </quer:query>
    </ExportQuery>
  </Content>
</Document>

```



It is mandatory to specify the attributes correctly.

Export Message

The following sample describes the export message.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/
envelope/"
  xmlns:xm="http://www.w3.org/2005/05/xmlmime"
  xmlns:add="http://www.w3.org/2005/03/addressing"
  xmlns:man="http://www.taleo.com/ws/integration/toolkit/2011/05/
management">
  <soapenv:Header>
    <add:MessageID>Message_id_101</add:MessageID>
    <add:Action>http://www.taleo.com/ws/integration/toolkit/2005/07/
action/
      export</add:Action>
    <add:ReplyTo>
      <add:Address>http://www.taleo.com/ws/integration/
toolkit/2005/07/
        addressing/queue</add:Address>
      <add:ReferenceParameters/>
    </add:ReplyTo>
  </soapenv:Header>
  <soapenv:Body>
    <man:submitLargeDocument>
      <Document xmlns="http://www.taleo.com/ws/integration/
toolkit/2011/05">
        <Attributes>
          <Attribute name="locale">en</Attribute>
          <Attribute name="largegraph">true</Attribute>
          <Attribute name="mode">T-XML</Attribute>
          <Attribute name="version">http://www.taleo.com/ws/
so800/
            2009/01</Attribute>
        </Attributes>
        <Content>
          <ExportQuery xmlns="http://www.taleo.com/ws/
so800/2009/01">
            <quer:query alias="Query752BAE88-55BD-11E2-8B1D-
CDAFDC756D0C"
              preventDuplicates="false"
              projectedClass="User"
              xmlns="http://www.taleo.com/ws/integration/
toolkit/2005/07"
              xmlns:m="http://www.taleo.com/ws/
integration/toolkit/2005/07"
              xmlns:SOAP-ENV="http://schemas.xmlsoap.org/
soap/envelope/"
              xmlns:tn="http://www.taleo.com/ws/
integration/tn/2005/07"
              xmlns:quer="http://www.taleo.com/ws/
integration/query"
              xmlns:wsa="http://www.w3.org/2005/03/
addressing">
              <quer:subQueries/>
              <quer:projections>
                <quer:projection>
                  <quer:field path="CorrespondenceEmail"/
                >
            >
          </ExportQuery>
        </Content>
      </Document>
    </man:submitLargeDocument>
  </soapenv:Body>
</soapenv:Envelope>
```

```

        </quer:projection>
        <quer:projection>
            <quer:field path="FirstName" />
        </quer:projection>
        <quer:projection>
            <quer:field path="LastName" />
        </quer:projection>
        <quer:projection>
            <quer:field path="WorkPhone" />
        </quer:projection>
    </quer:projections>
    <quer:projectionFilterings/>
    <quer:filterings>
        <quer:filtering>
            <quer:equal>
                <quer:field
path="CorrespondenceEmail" />
                <quer:string>test-
clientname@testorg.com</quer:string>
            </quer:equal>
        </quer:filtering>
    </quer:filterings>
    <quer:sortings/>
    <quer:sortingFilterings/>
    <quer:groupings/>
    <quer:joinings/>
</quer:query>
</ExportQuery>
</Content>
</Document>
</man:submitLargeDocument>
</soapenv:Body>
</soapenv:Envelope>

```

Designing a Data Extraction Process

When using Bulk API for data extraction, the following conditions should be considered:

- The frequency of the extractions.
- The average or expected size of the extractions.
- The type of communication with the customer.
- The complexity of the mapping required between the Taleo and customer formats.
- The type of data required from the Taleo application and its relation to the actual model.

Data Transformation

The system allows the option to transform and extract the data using SQ-XML; however, it is strongly suggested that the transformations are carried out only if:

1. The transformation is simple (date formatting, concatenations, simple DECODE functions), AND
2. The transformation can be completely expressed in SQ-XML (thus saving a customer specific integration package altogether).

If the second condition fails, then a customer specific package will have to exist anyway. It is thus optimal to group all the transformation information in it. One immediate gain is that the SQXML document will be much lighter and easier to read and maintain. If the first condition fails, then performance will degrade when executing the requests on the application and database servers.

Exceptions and Error Handling

• Exceptions and Error Handling.....	26
--------------------------------------	----

Exceptions and Error Handling

Overview

When integrating with Bulk API web service, there are three possible error conditions to consider. The primary error condition is the application service error. This condition typically arises when the service request has a business logic error, i.e. the import or export criteria in the request is incorrect. The web service will also result in an error if there is a technical or non-functional error such as a database failure, or a SOAP fault if SOAP XML is malformed. In Taleo Connect Server, the following error conditions are referred to:

1. Application Service Error – Caused by a business logic error in the request
2. Web Service Errors – Caused due to error in the SOAP
3. Unexpected Technical Errors – Caused by an unexpected system failure

Application Server Error

This error condition occurs when the web service request contains either invalid business logic or results in a technical exception. If it is a handled exception, Bulk API returns an error message along with a sub-code that provides the context of the error.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <ns1:getLargeDocumentByKeyResponse xmlns:ns1="http://www.taleo.com/
ws/integration/
  toolkit/2011/05/management">
      <Document xmlns="http://www.taleo.com/ws/integration/
  toolkit/2011/05">
        <Attributes>
          <Attribute name="count">0</Attribute>
          <Attribute name="duration">0:00:00.024</Attribute>
          <Attribute name="mode">T-XML</Attribute>
          <Attribute name="version">http://www.taleo.com/ws/itk/
prototype/2006/
          05/invalid</Attribute>
        </Attributes>
        <Content>
          <ExportErrors>
            <Error>
              <Code>ServerError</Code>
              <Subcode>itk:ExportInvalidParam</Subcode>
              <Reason>The attribute "version" value "http://
www.taleo.com/ws/
          itk/prototype/2006/05/invalid" is not supported
          by the
          export service.</Reason>
            </Error>
          </ExportErrors>
        </Content>
      </Document>
    </ns1:getLargeDocumentByKeyResponse>
  </soap:Body>
</soap:Envelope>
```

Web Service Errors

An error such as malformed SOAP package will result in Web Service error. This is managed by the web services framework and handled as a SOAP fault.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>Unexpected close tag </add:ReplyTo>; expected </
add:ReferenceParameters>.
        at [row,col {unknown-source}]: [11,19]</faultstring>
      <detail>
        <WebServiceFault xmlns="http://www.taleo.com/ws/integration/
toolkit/2005/07">
          <code>SystemError</code>
          <message>Unexpected close tag </add:ReplyTo>; expected </
add:ReferenceParameters>.
            at [row,col {unknown-source}]: [11,19]</message>
        </WebServiceFault>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

Unexpected Technical Errors

A technical error such as database failure, network error, JVM failure, or another system error will result in Unexpected Technical Error. Serialization of such error depends on the nature of the technical failure. An error resulting from unavailability of a resource may be serialized, however, depending on the nature of the failure (such as a database failure), the error may only be logged in the system log files.

Bulk API Security

• Bulk API Security.....	29
--------------------------	----

Bulk API Security

Bulk API offers the same levels of security for protecting data and logic as in Taleo user interfaces. This section describes the user role and authentication schemes required for accessing Bulk API.

User Roles

The zone can be accessed only after the user has been authenticated. To integrate with Taleo zone using Bulk API, a user with system integration role is required. The user account must be provisioned in the zone.

Supported Authentication Schemas

1. Bulk API currently supports Basic Authentication

Basic HTTP/S authentication is mandatory to access Taleo Web services.

MTOM Guide

- Using MTOM.....31

Using MTOM

SOAP Message Transmission Optimization Mechanism (MTOM) is a W3C recommendation designed for optimizing the electronic transmission of attachments. It improves memory management by streaming chunks of large attachment. Streaming the SOAP document as an attachment rather than transmitting it as base 64 encoded message, greatly reduces the memory utilization.

MTOM and Bulk API

The web service stack of Taleo Connect Server associates the `xs:base64Binary` type to a Java `DataSource` class. This class provides Java streaming capabilities to read the document content. Typically, the web service stack deserializes each element into an object in memory. But for the `base64Binary`, it provides a Data Source that has operations to access the content using streaming. The data is not read by the stack and therefore not automatically loaded in memory.

The stream is then read and written into temporary file storage (database). The content is therefore never entirely loaded in memory. This is particularly important in a cloud based environment where resources such as memory utilization must be carefully managed. The same concept is also applied for returning large content in a web service response.

Mandatory Use of MTOM

The two operations `submitLargeDocument` and `getLargeDocumentByKey` support transfer of large documents (size greater than 50kb) as attachments using MTOM. Taleo Connect Server will report an error if a SOAP payload greater than 50kb is sent or retrieved using any of the Bulk API operations. This limitation has been set to control the amount of memory used by Bulk API consumers to prevent any possible issue with memory usage.

The size of the SOAP being transferred is measured in entirety of the XML message. This means that the count starts with the first character of the XML message. This usually is the declaration of the xml version (eg.: `<?xml version='1.0'...`).

Operation Details

MTOM is required for the following two operations in the Bulk API:

1. `SubmitLargeDocument`
2. `GetLargeDocumentByKey`

SubmitLargeDocument()

The `tns:Document` parameter is of type `base64Binary` and requires MTOM for transmitting documents that are larger than 50KB.

Schema

```
<xsd:element name="submitLargeDocument">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="1"
        minOccurs="1"
        name="Document"
        nillable="true"
        type="nsm:StreamBody"
        nsxmlmime:expectedContentTypes="application/octet-
stream"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

getLargeDocumentByKey ()

The tns:Document return value is of type base64Binary. The format of the response is MTOM.

Schema

```
<xsd:element name="getLargeDocumentByKeyResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="1"
        minOccurs="1"
        name="Document"
        nillable="true"
        type="nsm:StreamBody"
        nsxmlmime:expectedContentTypes="application/octet-
stream"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Request Response Formats

There are no standards on how the response should be formatted for a request made using MTOM. The default implementation of the bulk API is as follows:

Operation	Request Format	Response Format
getLargeDocumentByKey	SOAP	MTOM
getMessageByKey	SOAP	SOAP
submitLargeDocument	MTOM	SOAP

Changing Request Response Formats

The Taleo Connect Server supports incoming request in the MTOM format for all operations.

Changing the response default format

Starting with release 13A, it is possible to override the default response format of an operation that is coded to return a response in the SOAP format. This is done by adding the ReturnAttachment element in the ReferenceParameters element of some of the following SOAP header elements:

- in "xmlns:soap='http://www.w3.org/2001/12/soap-envelope' soap:Header" element
- in "xmlns:wsa='http://www.w3.org/2005/03/addressing' wsa:ReplyTo" element
- in "xmlns:wsa='http://www.w3.org/2005/03/addressing' wsa:FaultTo" element

Section	Behavior
soap:Header	Set the ReturnAttachment element in this section to always receive a MTOM response (no matter if the operation is set to return in this format or not).
wsa:ReplyTo	Set the ReturnAttachment element in this section to receive a MTOM response when the operation completes with success.
wsa:FaultTo	Set the ReturnAttachment element in this section to receive a MTOM response when the operation completes in failure.

Example 1: Return success in MTOM format and failure in SOAP format.

```
<soap:Header>
  <wsa:ReplyTo>
    <wsa:ReferenceParameters xmlns:ns3="http://addressing.taleo.com/
ReturnAttachments">
      <ns3:ReturnAttachments>true</ns3:ReturnAttachments>
    </wsa:ReferenceParameters>
  </wsa:ReplyTo>
  <wsa:FaultTo>
    <wsa:ReferenceParameters xmlns:ns3="http://addressing.taleo.com/
ReturnAttachments">
      <ns3:ReturnAttachments>false</ns3:ReturnAttachments>
    </wsa:ReferenceParameters>
  </wsa:FaultTo>
</soap:Header>
```

Example 2: Example 1: Return in MTOM format whether success or failure.

```
<soap:Header>
  <wsa:ReferenceParameters xmlns:ns3="http://addressing.taleo.com/
ReturnAttachments">
    <ns3:ReturnAttachments>true</ns3:ReturnAttachments>
  </wsa:ReferenceParameters>
</soap:Header>
```

In the `IntegrationManagementService.wsdl`, the `ReturnAttachments` element is not explicitly defined under the `wsa:ReferenceParameters`. This cannot be done because the `wsa:ReferenceParameters` is a standard element of type 'ANY' defined in the W3C addressing schema (<http://www.w3.org/2005/03/addressing>). The definition of this element is:

```
<xs:element name="ReferenceParameters" type="tns:ReferenceParametersType"/>
  <xs:complexType name="ReferenceParametersType" mixed="false">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
```

The `ReturnAttachments` element is defined in its own schema that is included in the `IntegrationManagementService WSDL` file. This element namespace is `http://addressing.taleo.com/ReturnAttachments`. Here is a copy of the schema:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema
attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="http://addressing.taleo.com/ReturnAttachments">
```

```

    <xsd:annotation>
      <xsd:documentation>This element is used to specified if we want a
web service
      response in an attachment or not. The element needs to be
place in a
      ReferenceParameters element.</xsd:documentation>
    </xsd:annotation>
    <xsd:element name="ReturnAttachments" type="xsd:boolean"/>
</xsd:schema>

```

Microsoft .NET Streaming

In order to make the Bulk API WSDL compatible with the Microsoft .NET streaming technology, the standard base64Binary has been redefined as a new element defined in the Microsoft namespace.

```

<xsd:schema xmlns:tns=http://schemas.microsoft.com/Message
  xmlns:xs=http://www.w3.org/2001/XMLSchema
  elementFormDefault="qualified"
  targetNamespace="http://schemas.microsoft.com/Message">
  <xs:simpleType name="StreamBody">
    <xs:restriction base="xs:base64Binary"/>
  </xs:simpleType>
</xsd:schema>

```

MTOM Request Sample

Here is an example of a Candidate Merge request sent using the MTOM optimization. The important sections to note are:

- The <ns6:Document> element that includes the attachment
- The attachment itself in the part Content-Id equal to 51349b7a-562a-4e98-ab8c736cfecb03eb@example.jaxws.sun.com
- Note the attachment is not base64 encoded

```

POST /enterprise/soap?ServiceName=IntegrationManagementService HTTP/1.1
Proxy-Authorization: Content-type: multipart/related;
  start="<rootpart*7661941e-06d9-4961-af10-
db612c7ffcc4@example.jaxws.sun.com>";
  type="application/xop+xml";
  boundary="uuid:7661941e-06d9-4961-af10-db612c7ffcc4";
  start-info="text/xml"
Soapaction: "http://www.taleo.com/ws/integration/toolkit/2011/05/
management/
  IntegrationManagementService#submitLargeDocument"
Accept: text/xml, multipart/related, text/html, image/gif, image/jpeg, *;
  q=.2, */*; q=.2
User-Agent: Oracle JAX-WS 2.1.5
Transfer-Encoding: chunked
Host: intgfusion.qc.taleo.net
  --uuid:7661941e-06d9-4961-af10-db612c7ffcc4
Content-Id: <rootpart*7661941e-06d9-4961-af10-
db612c7ffcc4@example.jaxws.sun.com>
Content-Type: application/xop+xml;charset=utf-8;type="text/xml"
Content-Transfer-Encoding: binary
<?xml version='1.0' encoding='UTF-8'?>

```

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <ns2:MessageID xmlns:ns2="http://www.w3.org/2005/03/addressing"
      xmlns:ns3="http://www.taleo.com/ws/integration/
      toolkit/2005/07/action/import"
      xmlns:ns4="http://www.taleo.com/ws/integration/
      toolkit/2011/05"
      xmlns:ns5="http://www.taleo.com/ws/integration/
      toolkit/2005/07"
      xmlns:ns6="http://www.taleo.com/ws/integration/
      toolkit/2011/05/management"
      xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
      95dad2b7-4a38-4b1a-a862-4f69aa10c06c</ns2:MessageID>
    <ns2:ReplyTo xmlns:ns2="http://www.w3.org/2005/03/addressing"
      xmlns:ns3="http://www.taleo.com/ws/integration/
      toolkit/2005/07/action/import"
      xmlns:ns4="http://www.taleo.com/ws/integration/
      toolkit/2011/05"
      xmlns:ns5="http://www.taleo.com/ws/integration/
      toolkit/2005/07"
      xmlns:ns6="http://www.taleo.com/ws/integration/
      toolkit/2011/05/management"
      xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
      <ns2:Address>http://www.taleo.com/ws/integration/
      toolkit/2005/07/addressing/
      queue</ns2:Address>
    </ns2:ReplyTo>
    <ns2:Action xmlns:ns2="http://www.w3.org/2005/03/addressing"
      xmlns:ns3="http://www.taleo.com/ws/integration/
      toolkit/2005/07/action/import"
      xmlns:ns4="http://www.taleo.com/ws/integration/
      toolkit/2011/05"
      xmlns:ns5="http://www.taleo.com/ws/integration/
      toolkit/2005/07"
      xmlns:ns6="http://www.taleo.com/ws/integration/
      toolkit/2011/05/management"
      xmlns:xmime="http://www.w3.org/2005/05/xmlmime">http://
      www.taleo.com/ws/
      integration/toolkit/2005/07/action/import
    </ns2:Action>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/
      oasis-200401-wss-wssecurity-secext-1.0.xsd"
      S:mustUnderstand="1">
      <wsse:UsernameToken xmlns:wsse="http://docs.oasis-open.org/
      wss/2004/01/
      oasis-200401-wss-wssecurity-secext-1.0.xsd"
      xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
      oasis-200401-wss-wssecurity-utility-1.0.xsd"
      wsu:Id="UsernameToken-9HkgL4aMC7m4RQiPY0z22Q22">
        <wsse:Username>theUser</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/
      wss/2004/01/
      oasis-200401-wss-username-token-
      profile-1.0#PasswordText">thePassword
        </wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </S:Header>
</S:Body>

```

```

        <ns6:submitLargeDocument xmlns:ns2="http://www.w3.org/2005/03/
addressing"
            xmlns:ns3="http://www.taleo.com/ws/integration/
toolkit/2005/07/action/import"
            xmlns:ns4="http://www.taleo.com/ws/integration/
toolkit/2011/05"
            xmlns:ns5="http://www.taleo.com/ws/integration/
toolkit/2005/07"
            xmlns:ns6="http://www.taleo.com/ws/integration/
toolkit/2011/05/management"
            xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
        <ns6:Document>
            <Include xmlns="http://www.w3.org/2004/08/xop/include"
                href="cid:51349b7a-562a-4e98-
ab8c-736cfecb03eb@example.jaxws.sun.com"/>
            </ns6:Document>
        </ns6:submitLargeDocument>
    </S:Body>
</S:Envelope>
--uuid:7661941e-06d9-4961-af10-db612c7ffcc4
Content-Id: <51349b7a-562a-4e98-ab8c-736cfecb03eb@example.jaxws.sun.com>
Content-Type: text/xml
Content-Transfer-Encoding: binary

<ImportEntities>
    <Candidate-merge xmlns="http://www.taleo.com/ws/tee800/2009/01">
        <Candidate>
            <EmailAddress searchType="searchAndValue"
                searchTarget="."
                searchValue="tmerge_001@invalidemail.com">
                tmerge_001@invalidemail.com
            </EmailAddress>
            <FirstName>tmerge_fn</FirstName>
            <MiddleInitial>tmerge_mi</MiddleInitial>
            <LastName>tmerge_ln</LastName>
        </Candidate>
    </Candidate-merge>
</ImportEntities>
--uuid:7661941e-06d9-4961-af10-db612c7ffcc4--

```

MTOM Response Sample

Here is an example of a New Hire Export response sent using the MTOM. The important sections to look at are:

- The <ns1:Document> element that includes the attachment
- The attachment itself in the part with the Content-Id equal to 13521516291925942006291584@http://www.w3.org/2001/XMLSchema
- Note the attachment is not base64 encoded

```

HTTP/1.1 200 OK
Date: Mon, 05 Nov 2012 21:40:28 GMT
Server: Taleo Web Server 8
Vary: Accept-Encoding
Content-Length: 18149
Content-Type: multipart/related; type="application/xop+xml";
    start="<soap.xml@xfire.codehaus.org>"; start-info="text/xml";
    boundary="-----_Part_80_1296222611.1352151629192"

```



```

-----_Part_80_1296222611.1352151629192
Content-Type: application/xop+xml; charset=UTF-8; type="text/xml"
Content-Transfer-Encoding: 8bit
Content-ID: <soap.xml@xfire.codehaus.org>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <ns1:getLargeDocumentByKeyResponse
      xmlns:ns1="http://www.taleo.com/ws/integration/
        toolkit/2011/05/management">
      <ns1:Document xmlns:ns2="http://www.w3.org/2004/11/xmlmime"
        ns2:contentType="text/xml; charset=UTF-8">
        <Include xmlns="http://www.w3.org/2004/08/xop/include"
          href="cid:13521516291925942006291584@http://
            www.w3.org/2001/XMLSchema"/>
        </ns1:Document>
      </ns1:getLargeDocumentByKeyResponse>
    </soap:Body>
  </soap:Envelope>
-----_Part_80_1296222611.1352151629192
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: binary
Content-ID: <13521516291925942006291584@http://www.w3.org/2001/XMLSchema>

<Document xmlns="http://www.taleo.com/ws/integration/toolkit/2011/05">
  <Attributes>
    <Attribute name="duration">0:00:04.145</Attribute>
    <Attribute name="count">8</Attribute>
    <Attribute name="entity">Application</Attribute>
    <Attribute name="mode">T-XML</Attribute>
    <Attribute name="version">http://www.taleo.com/ws/tee800/2009/01</
  Attribute>
  </Attributes>
  <Content>
    <ExportTXML xmlns="http://www.taleo.com/ws/integration/
      toolkit/2005/07/action/export"
      xmlns:e="http://www.taleo.com/ws/tee800/2009/01">
      <e:Application>
        <e:CSWLatestDate>2005-10-05T23:30:22-04:00</
      e:CSWLatestDate>
        <e:Candidate>
          <e:Candidate>
            <e:Address>330 St-Vallier Est</e:Address>
            <e:Address2>Suite 400</e:Address2>
            <e:Birthday/>
            <e:City>Quebec</e:City>
            <e:Credentials>
              <e:Credentials>
                <e:Username>nhe_candidat1@invalidemail.com</e:Username>
                </e:Credentials>
              </e:Credentials>
              <e:EmailAddress>nhe_candidat1@invalidemail.com</
            e:EmailAddress>
              <e:EmployeeNumber>NHE_Candidat1</e:EmployeeNumber>
              <e:FirstName>NHE_Candidat1</e:FirstName>
              <e:InternalApplication>true</e:InternalApplication>
              <e:LastName>NHE_Candidat1</e:LastName>
            </e:Credentials>
          </e:Candidate>
        </e:Candidate>
      </e:Application>
    </ExportTXML>
  </Content>
</Document>

```

```

        <e:MiddleInitial/>
        <e:Number>5165</e:Number>
        <e:SocialSecurityNumber>123 456 789</
e:SocialSecurityNumber>
        <e:ZipCode>G1K 9C5</e:ZipCode>
    </e:Candidate>
</e:Candidate>
<e:Offers>
    <e:Offer>
        <e:ActualStartDate>2003-10-27T00:00:00-05:00</
e:ActualStartDate>
        <e:AnnualBonus>
            <e:value locale="en">Y</e:value>
        </e:AnnualBonus>
        <e:CarAllowance>
            <e:value locale="en"/>
        </e:CarAllowance>
        <e:RelocationAmount/>
        <e:RelocationType></e:RelocationType>
        <e:Salary/>
        <e:SignOnBonus>2000.0</e:SignOnBonus>
        <e:StockOption>
            <e:value locale="en">Y</e:value>
        </e:StockOption>
        <e:StockOptionType></e:StockOptionType>
        <e:StockType></e:StockType>
        <e:Vacation>15</e:Vacation>
        <e:VacationType>
            <e:VacationType>
                <e:Description>
                    <e:value locale="en">Weeks</e:value>
                </e:Description>
                <e:Number>1</e:Number>
            </e:VacationType>
        </e:VacationType>
    </e:Offer>
</e:Offers>
<e:Requisition>
    <e:Requisition>
        <e:ContestNumber>IntOfferNewHire-001</
e:ContestNumber>
        <e:Department></e:Department>
        <e:JobInformation>
            <e:JobInformation>
                <e:RecruiterOwner>
                    <e:User>
                        <e:FirstName>Christian</
e:FirstName>
                            <e:LastName>Allard</e:LastName>
                        </e:User>
                    </e:RecruiterOwner>
                </e:JobInformation>
            </e:JobInformation>
            <e:Position></e:Position>
        </e:Requisition>
    </e:Requisition>
</e:Application>
<e:Application>
    <e:CSWLatestDate>2005-10-05T23:30:21-04:00</
e:CSWLatestDate>
    <e:Candidate>

```

```

    <e:Candidate>
      <e:Address>330 St-Vallier Est</e:Address>
      <e:Address2>Suite 400</e:Address2>
      <e:Birthday/>
      <e:City>Quebec</e:City>
      <e:Credentials>
        <e:Credentials>

<e:Username>nhe_candidat3@invalidemail.com</e:Username>
      </e:Credentials>
    </e:Credentials>
    <e:EmailAddress>nhe_candidat3@invalidemail.com</
e:EmailAddress>

    <e:EmployeeNumber>NHE_Candidat3</e:EmployeeNumber>
    <e:FirstName>NHE_Candidat3</e:FirstName>
    <e:InternalApplication>true</e:InternalApplication>
    <e:LastName>NHE_Candidat3</e:LastName>
    <e:MiddleInitial/>
    <e:Number>5168</e:Number>
    <e:SocialSecurityNumber>123 456 789</
e:SocialSecurityNumber>
      <e:ZipCode>G1K 9C5</e:ZipCode>
    </e:Candidate>
  </e:Candidate>
  <e:Offers>
    <e:Offer>
      <e:ActualStartDate>2003-10-27T00:00:00-05:00</
e:ActualStartDate>

      <e:AnnualBonus>
        <e:value locale="en">Y</e:value>
      </e:AnnualBonus>
      <e:CarAllowance>
        <e:value locale="en"/>
      </e:CarAllowance>
      <e:RelocationAmount/>
      <e:RelocationType></e:RelocationType>
      <e:Salary/>
      <e:SignOnBonus>2000.0</e:SignOnBonus>
      <e:StockOption>
        <e:value locale="en">Y</e:value>
      </e:StockOption>
      <e:StockOptionType></e:StockOptionType>
      <e:StockType></e:StockType>
      <e:Vacation>15</e:Vacation>
      <e:VacationType>
        <e:VacationType>
          <e:Description>
            <e:value locale="en">Weeks</e:value>
          </e:Description>
          <e:Number>1</e:Number>
        </e:VacationType>
      </e:VacationType>
    </e:Offer>
  </e:Offers>
  <e:Requisition>
    <e:Requisition>
      <e:ContestNumber>IntOfferNewHire-001</
e:ContestNumber>

      <e:Department></e:Department>
      <e:JobInformation>
        <e:JobInformation>

```

```

                                <e:RecruiterOwner>
                                  <e:User>
                                    <e:FirstName>Christian</
e:FirstName>
                                    <e:LastName>Allard</e:LastName>
                                  </e:User>
                                </e:RecruiterOwner>
                              </e:JobInformation>
                            </e:JobInformation>
                          <e:Position></e:Position>
                        </e:Requisition>
                      </e:Requisition>
                    </e:Application>
                  </ExportTXML>
                </Content>
              </Document>
-----=_Part_80_1296222611.1352151629192--

```

Troubleshooting Guide

- Troubleshooting Overview..... 42

Troubleshooting Overview

This section describes the options available to troubleshoot Bulk API calls. Taleo integration platform provides two options to review the request and response XMLs as well as the server side error logs. For testing basic connectivity and zone/user access, a basic SoapUI project is available along with this documentation.

Request/Response XMLs in TCC

Using Taleo Connect Client (TCC), the request and response XMLs can be easily viewed. This tool can be used for verifying correctness of the XML document contained in the request as well as the error message or reason in the response. Here are the steps to set-up the tools:

Prerequisite

TCC must be installed and you must be logged in to your zone.

Steps

1. In the file menu, click New Profile Wizard.
The New Profile Wizard dialog opens.
2. Complete the Profile type selection. Click Next.
3. Click Finish.

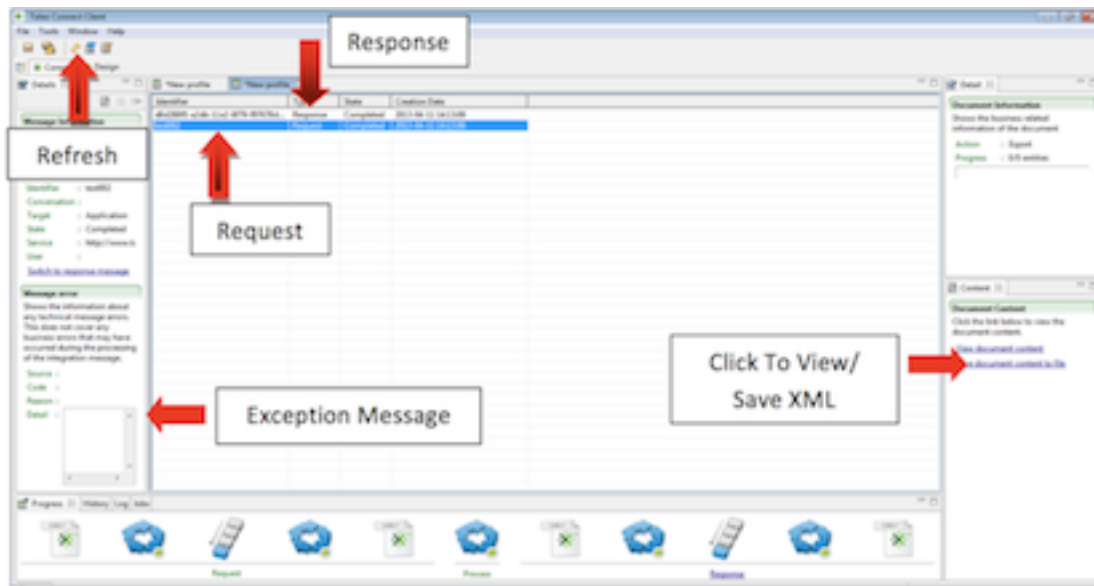
Result

From the resulting console, the following can be viewed:

- Bulk API request XML – based on the message-id provided by the calling process
- Bulk API response XML
- Exception message

Logging will be available in TCC only when the request has been correctly processed by the ITK and persisted in the ITK queue. If something goes wrong before the request is received by the Integration Tool Kit, the error message won't be available in TCC.

The relevant sections of the console are indicated below (for more information, see the Taleo Connect User Guide):



Downloading Error Logs

The error logs can be downloaded from the zone and are available as .zip files.

Prerequisite

Log on to the Taleo zone with support user credentials

Excessive logging can impact performance of the zone. This option is available only for use during development and system integration phases; it is not available in production.

Steps

1. Go to *Configuration > SmartOrg > Administration > [Production Security] > System Info*
2. In the Quick Access section on the left, click Log4j.
3. In the Appender Name column, search for "WEBSERVICES_ACTIVITY" – this will be towards page 4
4. Set the Status Level for "WEBSERVICES_ACTIVITY" to required logging level.
5. Call web service operations.

All	Has the lowest possible rank and is intended to turn on all logging.
DEBUG	Designates fine-grained informational events that are most useful to debug an application.
ERROR	Designates error events that might still allow the application to continue running.
FATAL	Designates very severe error events that will presumably lead the application to abort.
INFO	Designates informational messages that highlight the progress of the application at coarse-grained level.
OFF	Has the highest possible rank and is intended to turn off logging.

TRACE	Designates finer-grained informational events than the DEBUG.
TRACE_INT	Designates TRACE level integer value.
WARN	Designates potentially harmful situations.

The logs created are available for download.

6. Click Download all.

Result

The error logs for each product are available on their respective sites:

- [https://\[yourzone\]/smartorg/akirax/admin/loggerUtil.jsf](https://[yourzone]/smartorg/akirax/admin/loggerUtil.jsf)
- [https://\[yourzone\]/enterprise/akirax/admin/loggerUtil.jsf](https://[yourzone]/enterprise/akirax/admin/loggerUtil.jsf)
- [https://\[yourzone\]/orion/akirax/admin/loggerUtil.jsf](https://[yourzone]/orion/akirax/admin/loggerUtil.jsf)
- [https://\[yourzone\]/onboarding/akirax/admin/loggerUtil.jsf](https://[yourzone]/onboarding/akirax/admin/loggerUtil.jsf)

Appendix

• Volume Limits.....	46
• List of Possible Errors.....	47
• Taleo Connect Server.....	54

Volume Limits

The volume limits specifies maximums for the number of records per day or transaction for Bulk API and TCC integration.

Basic Offering

All Customers using Taleo Connect Client and Taleo Managed Integrations immediately get access to the basic offering and are subject to the following volume limits:

- Maximum of 500 000 extracted records per day.
- Maximum of 100 000 records per extraction transaction.
- Maximum of 50 000 imported records per day.
- Unrestricted number of records per import transaction.

Premium Offering

Customers with the premium offering are subject to the following volume limits:

- Maximum of 5 000 000 extracted records per day.
- Maximum of 300 000 records per extraction transaction.
- Maximum of 350 000 imported records per day.
- Unrestricted number of records per import transaction.

Planned Exception Framework

Where a project is being undertaken and volume limits need to be modified to accommodate temporary changes, a planned exception can be requested.

Taleo recognizes the following types of projects with corresponding volume limits.

Project Type	Daily Import	Daily Export	Per Transaction Export
Data Migrations (New/V2T)	350 000*	250 000	100 000, 256 MB
Data Migrations (Acquisitions/Expansion) – OLTP users	150 000*	250 000	100 000, 256 MB
Data warehouse initial extract – OLTP users	150 000*	10 million	100 000, 256 MB
Final Extract (non- Oracle dump)	150 000*	10 million	100 000, 256 MB
Large scale data refresh or modifications**	150 000*	250 000	100 000, 512 MB

* Can go up to 350K but requires signing waiver of liability of SLA conformance risk with Taleo. Taleo also reserves the right to disallow a higher limit if there are indications that this has impact on other customers running on same hardware.

** This is a temporary increase only, not to exceed a two-week period. No more than three planned and unplanned exceptions may be requested under this project type category per year.

List of Possible Errors

Possible Errors List

Empty key

- Category: Request
- Error: The provided key is empty or null.
- Cause: No key has been specified to retrieve the message.
- Action: Specify a valid key.

Invalid key

- Category: Request
- Error: No message matches the provided key '{key}'.
- Cause: An invalid key has been specified to retrieve the message.
- Action: Specify a valid key.

Following is an error specific to `getDocumentByKey`, refer to `getLargeDocumentByKey(Retrieve)` section above for errors that are shared by both operations.

50k limit for response document

- Category: System
- Error: The memory response size threshold has been reached. Memory Threshold value: {threshold_value}, Actual read size: {current_value}. Use the `getLargeDocumentByKey` with attachment instead of `getDocumentByKey`.
- Cause: The response file exceed the memory threshold allowed for a web service call
- Action: Use the `getLargeDocumentByKey` with attachment instead of `getDocumentByKey`.

This section will list general errors that can occur while invoking any type of Taleo web services. Additional errors about specific web services will be described later in this document.

Empty SOAP body

- Category: Request
- Error: The service '{service.name}' is not registered for the provided version ({namespace.list}) in the XML request for the current product ({application.code}). This can also be caused by an empty body part.
- Cause: The request is incomplete.
- Action: Make sure to create a complete request respecting the web service WSDL.

Invalid password

- Category: System
- Error: AccessResult: result: Access Denied | AuthenticationAsked: true | ErrorCode: LOGIN_ERROR_INVALID_PASSWORD: 2 | ErrorReason: null
- Cause: Invalid password has been provided.
- Action: Provide valid password.

Invalid username

- Category: System
- Error: AccessResult: result: Access Denied | AuthenticationAsked: true | ErrorCode: LOGIN_ERROR_INVALID_USERNAME: 1 | ErrorReason: null
- Cause: Invalid username has been provided.

- Action: Provide valid username.

Malformed xml document

- Category: Request
- Error: A problem occurred while deserializing the parameter reference at the position index 0.
- Cause: The provided XML document is not well formed.
- Action: Make sure the XML document is well formed.

Malformed xml document in attachment

- Category: Request
- Error: The provided XML in the SOAP attachment is not well formed.
- Cause: The provided XML document in the SOAP attachment is not well formed.
- Action: Make sure the XML document is well formed according to the offering schema.

Missing SOAP body

- Category: Request
- Error: A parsing error has been encountered. One possible reason is that the body element is missing from the request.
- Cause: The request is incomplete
- Action: Make sure to create a complete request.

Unknown SOAP version

- Category: Request
- Error: Invalid SOAP version: {soap.version}
- Cause: The provided SOAP version is invalid.
- Action: Use a valid SOAP version.

Empty key

- Category: Request
- Error: The provided key is empty or null.
- Cause: No key has been specified to retrieve the message.
- Action: Specify a valid key.

Invalid key

- Category: Request
- Error: No message matches the provided key '{key}'.
- Cause: An invalid key has been specified to retrieve the message.
- Action: Specify a valid key.

1228800k limit for input document

- Category: System
- Error: The memory request size threshold has been reached. Memory Threshold value: {Max_threshold_value}, Actual read size: {current_value}. Document exceeding maximum allowed size ({Max_threshold_value}) of attachment.
- Cause: The request file must be split into multiple part.
- Action: Use the submitLargeDocument instead of submitDocument.

SOAP header with invalid Action value

- Category: Request
- Error: The requested Action is not available: {action}
- Cause: The provided Action element is invalid.

- Action: Set a valid Action element in the SOAP header. Possible values are:
 - <http://www.taleo.com/ws/integration/toolkit/2005/07/action/import>
 - <http://www.taleo.com/ws/integration/toolkit/2005/07/action/export>

SOAP header with invalid Address value

- Category: Request
- Error: The provided Address value {value} in the {element.path} element is invalid.
- Cause: The WS-Addressing endpoint reference requires a valid Address element.
- Action: Set a valid Address. Possible value is:
 - <http://www.taleo.com/ws/integration/toolkit/2005/07/addressing/queue>

SOAP header with missing Action element or empty value

- Category: Request
- Error: An Action element in the SOAP header section is missing or cannot be empty.
- Cause: The Action element in the SOAP header is missing or it is empty.
- Action: Set a valid Action element in the SOAP header. Possible values are:
 - <http://www.taleo.com/ws/integration/toolkit/2005/07/action/import>
 - <http://www.taleo.com/ws/integration/toolkit/2005/07/action/export>

SOAP header with missing Address element or empty

- Category: Request
- Error: The "{element.path}" is missing the mandatory "Address" element or it cannot be empty.
- Cause: The WS-Addressing endpoint reference requires a valid Address element.
- Action: Set a valid Address. Possible value is:
 - <http://www.taleo.com/ws/integration/toolkit/2005/07/addressing/queue>

SOAP header with missing MessageID element or empty value

- Category: Request
- Error: A MessageID element in the SOAP header section is missing or cannot be empty.
- Cause: The MessageID element in the SOAP header is missing or it is not empty.
- Action: Set a value in the MessageID element.

SOAP header with missing ReplyTo element or empty value

- Category: Request
- Error: The {element.path} is missing the mandatory "ReplyTo" element or it cannot be empty.
- Cause: The ReplyTo in the SOAP header is missing or it is empty.
- Action: Set a value in the ReplyTo element.

Unsupported keyword for an action

- Category: System
- Error: The {keyword} keyword is unsupported by the action {action}.
- Cause: The action does not support the provided keyword.
- Action: Verify the action used and/or validate the keyword.

Following are errors specific to submitDocument, refer to submitLargeDocument(Send) section below for errors that are shared by both operations.

50k limit for Base64 document

- Category: System

- Error: The memory request size threshold has been reached. Memory Threshold value: {threshold_value}, Actual read size: {current_value}. Use the submitLargeDocument instead of submitDocument.
- Cause: The request Base64 file exceed the memory threshold allowed for a web service call
- Action: Use the submitLargeDocument instead of submitDocument.

50k limit for input document

- Category: System
- Error: The memory request size threshold has been reached. Memory Threshold value: {threshold_value}, Actual read size: {current_value}. Use the submitLargeDocument instead of submitDocument.
- Cause: The request file exceed the memory threshold allowed for a web service call
- Action: Use the submitLargeDocument instead of submitDocument.

Entity not found within relation

- Category: Application
- Error: The specified entity cannot be found. Path: {path}. Search criteria: {search.criteria}
- Cause: The searched entity does not exist.
- Action: Create the appropriate entity before referencing it or fix the search value to match already existing data.

Import daily limit reached

- Category: System
- Error: The allowed number of daily imported entities ({max.count}) has been reached ({current.count}).
- Cause: The number of imported entities has been reached for the day.
- Action: Reduce the number of entities to import or contact support to increase the number of records that can be imported.

Invalid element within multilingual

- Category: Request
- The "value" element is expected instead of "{current.element}" for path "{path}".
- Action: Correct the information regarding the multilingual field. It must have the following structure:

```
<Description>
  <value locale="en">{description}</value>
</Description>
```

Invalid entity, path or field name in import request

- Category: Request
- Error: {entity}.{path}.{field} is not a valid entity, path or field for the version '{namespace.datamodel}' or the required access type for this operation cannot be satisfied.
- Cause: One of the provided entity, path or field does not exist in the data model version.
- Action: Validate the XML against the product data dictionary.

Invalid value for field

- Category: Application
- Error: Value {value} is invalid for field {path.field}.
- Cause: An invalid value has been entered for this field.
- Action: Provide valid value.

Missing or invalid locale with multilingual

- Category: Request

- Error: Locale "{current.locale}" for path "{path}" is invalid.
- Cause: A multilingual field is defined with an empty or invalid locale.
- Action: Set a valid locale to the multilingual field.

Missing value element with multilingual

- Category: Request
- Error: Missing "value" element for multilingual field "{path}".
- Cause: A multilingual field is specified without any value element and locale information.
- Action: Correct the information regarding the multilingual field. It must have the following structure:

```
<Description>
  <value locale="en">{description}</value>
</Description>
```

Invalid preparator class

- Category: Request
- Error: The supplied class ({value}) in the "preparator.class" document attribute does not implement the DocumentProcessingPreparator interface for the current product ({application.code}).
- Cause: The class specified is invalid for the current product.
- Action: Supply a valid class for the preparator.

Multilingual information for unilingual field

- Category: Request
- Error: The "value" element cannot be used for a unilingual field. Path: "{path}".
- Cause: A unilingual field contains locale information.
- Action: Remove the information regarding the locale.

Remove element from 1-1 relation

- Category: Application
- Error: The remove action is not supported on relations that require one and only one element. Path: {class}
- Cause: It is not possible to remove the element from a 1-1 relation.
- Action: Eliminate the remove action for that relation.

Unknown preparator class

- Category: Request
- Error: The supplied class ({value}) in the "preparator.class" document attribute has not been found using the default class loader for the current product ({application.code}).
- Cause: The class specified is not known for the current product.
- Action: Supply a valid class for the preparator.

Unknown preparator identifier

- Category: Request
- Error: The supplied identifier ({value}) in the "preparator" document attribute is not known for the current product ({application.code}).
- Cause: The identifier specified is not known for the current product.
- Action: Supply a valid identifier for the preparator.

Custom function contains invalid character

- Category: Application

- Error: Invalid customFunction: "{function}". Only letters and underscores are allowed.
- Cause: The specified customFunction name is invalid. Only letters and underscores are allowed.
- Action: Put a valid customFunction name.

Custom value contains invalid character

- Category: Application
- Error: Invalid value "{value}" for "expression" element in a "customValue" element. The value must refer to a column name.
- Cause: The value in the expression is invalid.
- Action: Put a valid value in the expression element.

Duplicate use of allFields projection

- Category: System
- Error: The allFields projection is exclusive, no other projections are allowed.
- Cause: Used projection allFields more than once for the same entity.
- Action: Do not use allFields projection more than once per entity.

Export daily limit reached

- Category: System
- Error: The allowed number of daily exported entities ({max.count}) has been reached ({current.count}).
- Cause: The number of exported entities has been reached for the day.
- Action: Change the query to export less entity or contact support to increase this limit.

Export not fast enough

- Category: System
- Error: The current number of entities exported per hour ({current.rate}) is below the minimum allowed rate ({min.rate}).
- Cause: The export rate per hour for the entities is too low.
- Action: Query must be optimized.

Export query execution error

- Category: Application
- Error: An error occurred during the query execution: {ORA error}.
- Cause: A database error occurred when executing the export request.
- Action: Consult the Oracle documentation to get more information about the reported ORA error.

Export size too large

- Category: System
- Error: The processing of this export request has generated a result whose size has exceeded the maximum allowed size ({max.size} bytes).
- Cause: The result size has exceeded the maximum allowed size.
- Action: Adjust the filtering to be within the maximum allowed size of the export result file.

Export time too long

- Category: System
- Error: The processing for this export request has exceeded the maximum allowed time.
- Cause: The processing for the export exceed the maximum allowed time.
- Action: Query must be optimized.

Export too many records

- Category: System

- Error: The number of entities returned by this export request ({current.count}) has exceeded the maximum allowed ({max.count}).
- Cause: The number of entities returned by this export request has exceeded the maximum allowed.
- Action: Adjust the filtering to be within the maximum number of records or contact support to increase the number of records that can be exported.

Invalid entity name in export request

- Category: Request
- Error: Unknown field or relation: "{field.name}" for the version: {datamodel}
- Cause: One of the provided entities does not exist in the data model version.
- Action: Validate the XML against the product data dictionary.

Invalid path or field name in export request

- Category: Request
- Error: Unknown entity: "{field.name}" for the version: {datamodel}
- Cause: One of the provided paths or fields does not exist in the data model version.
- Action: Validate the XML against the product data dictionary.

Maximum filtering count reached

- Category: Request
- Error: Query does not conform to performance metric requirements. Attribute "perfo.max.filtering.field.count" is {max.count} but the provided query has {current.count} occurrences.
- Cause: The number of filtering count was higher than the limit set by the attribute.
- Action: Remove filtering to be within the limit or contact support to increase the maximum filtering allowed.

Maximum projection count reached

- Category: Request
- Error: Query does not conform to performance metric requirements. Attribute "perfo.max.projection.field.count" is {max.count} but the provided query has {current.count} occurrences.
- Cause: The number of projection count was higher than the limit set by the attribute.
- Action: Remove projections to be within the limit or contact support to increase the maximum projection allowed.

Missing mandatory parameter

- Category: Request
- Error: The missing parameter "{attribute}" is mandatory for the export service.
- Cause: The value provided in the attribute cannot be empty.
- Action: Make sure to specify a valid value for the attribute.

Unsupported integration model version Export

- Category: Request
- Error: The attribute "{attribute.name}" value "{data.model.version}" is not supported by the export service.
- Cause: The data model version provided in the version attribute is not supported by the export service.
- Action: Validate the value provided in the version attribute. The model version of the product is available in the product data dictionary under the **Data Model Version** section.

Use of deprecated export operations

- Category: System
- Error: Forbidden element: {element.name}
- Cause: This element is not allowed.
- Action: Contact customer support.

Taleo Connect Server

Message Processing

Name	XML Tag	Description
Message Number	<MessageKey>	A number that identifies a message instance for internal purposes and that is unique for a given Taleo Connect Server instance within a host application.
Message Target	<Target>	A flag indicating how the asynchronous controller should process this message. Possible value is QUEUE (outbound) = 4.
Message State	<State>	The current state of the message. Possible values are NEW=1, INCOMING=2, READY=3, INPROGRESS=4, COMPLETED=5, INTERRUPTED=6, SUSPENDING=7, SUSPENDED=8, INERROR=9, STOPPING=10.
Message Format	<Format>	The format to use for the message envelope. Possible values are SOAP11=2 and SOAP12=3.
Correlation Key	<CorrelationKey>	An optional reference to another message number that is created for message instances participating in a request-response Message Exchange Pattern.
Conversation Identifier	<ConversationID>	The conversation identifier to which this message is linked. This value has no business significance in this version of Taleo Connect Server.

Description of Web Service Parameters

Name	Description
wsa:MessageID	This value is used to identify the message and to correlate it with its corresponding response message.
wsa:ReplyTo	This value indicates how Taleo Connect Server is to process the included document.
wsa:Action	This value indicates how Taleo Connect Server is to return the corresponding response message. Currently, only one value is allowed: http://www.taleo.com/ws/integration/toolkit/2005/07/addressing/queue
tns:Document	This element contains either a base64 encoded s2011_05:Document or a reference to a MTOM attachment. Type: nsm:StreamBody
ns2011_05:Document	A document is a generic container that holds the information elements of the message.

Name	Description
	Type: Complex Type
ns2011_05:Document.Attributes	The attributes represent document level information elements. Type: Complex Type
ns2011_05:Document.Attributes.Attribute	An attribute is a document level information element. Type: Complex Type
ns2011_05:Document.Attributes.Attribute.name	The attribute name uniquely identifies it among the attribute collections of a document. Type: xsd:string
ns2011_05:Document.Content	The document content represents the record level information elements. This type of information in the document content is dependent on the processing context. As specified by the schema, the Content element always contains exactly one child element which is the root of the content. Each immediate child of this element is considered a record in the Taleo Connect terminology. Type: xsd:any

States

Taleo Connect Server sets the state to **NEW** for any incoming message. While the document content is streamed, the state is set to **INCOMING**. When the complete document has been received, the state is set to **READY**. Any message present in the Taleo Connect Server queue in a **READY** state is eligible for processing by the asynchronous controller. The outbound requests are given a higher priority to the inbound requests. The inbound and outbound requests are processed in a **FIFO** order. Once the request has been processed, the asynchronous controller sets the message state to **COMPLETED** and the document is available for retrieval by a subsequent web service call.

Name	Description
NEW	A new message instance is created. State=1.
INCOMING	During the creation of the message instance content. State=2.
READY	The message instance is ready for processing by the asynchronous controller. State=3.
INPROGRESS	The message is currently being processed by the asynchronous controller. State=4.
COMPLETED	The asynchronous controller has completed the processing of the message instance. State=5.
INTERRUPTED	The message instance was being processed by the asynchronous controller, but the execution was not completed normally. State=6.
SUSPENDING	An external party has requested that the processing of this message instance be suspended. Can only occur if the message instance state is INPROGRESS . State=7.
SUSPENDED	The asynchronous controller has acknowledged the suspension request for this message instance. Can only occur if the message instance state is SUSPENDING . State=8.

Name	Description
INERROR	An unrecoverable error occurred while receiving the message. State=9.
STOPPING	The Taleo product has requested an immediate halt of this message processing for reasons independent of the Taleo Connect Server. State=10.

Namespaces

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:nsxmimime="http://www.w3.org/2005/05/xmlmime" attributeFormDefault="unqualified"
elementFormDefault="qualified" targetNamespace="http://www.taleo.com/ws/integration/toolkit/2011/05/management">
<xsd:import namespace="http://www.taleo.com/ws/integration/toolkit/2011/05" />
<xsd:import namespace="http://www.taleo.com/ws/integration/toolkit/2005/07/action/import" />
<xsd:import namespace="http://www.taleo.com/ws/integration/toolkit/2005/07/action/export" />
<xsd:import namespace="http://www.w3.org/2005/05/xmlmime" schemaLocation="http://www.w3.org/2005/05/xmlmime" />
<xsd:import namespace="http://addressing.taleo.com/ReturnAttachments" />
<xsd:import namespace="http://schemas.microsoft.com/Message" />
```

For .NET Streaming

```
<xsd:schema xmlns:tns="http://schemas.microsoft.com/Message" xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/Message">
- <xs:simpleType name="StreamBody">
<xs:restriction base="xs:base64Binary" />
</xs:simpleType>
</xsd:schema>
```

Prefix	Names	Usage
ai	http://www.taleo.com/ws/integration/toolkit/2005/07/action/import	Import result document structure.
nsxmimime	http://www.w3.org/2005/05/xmlmime	Mime type of attachments used to transport large content
ns2005_07	http://www.taleo.com/ws/integration/toolkit/2005/07	ITK namespace containing the standard definition of web service fault.
ns2011_05	http://www.taleo.com/ws/integration/toolkit/2011/05	ITK namespace containing the definition of the bulk API message structure
nsm	http://schemas.microsoft.com/Message	Required to be compatible with .NET C# streaming
tns	http://www.taleo.com/ws/integration/toolkit/2011/05/management	Integration Management Service Namespace
wsa	http://www.w3.org/2005/03/addressing	WS-Addressing
wsdlssoap	http://schemas.xmlsoap.org/wsdl/soap/	SOAP Binding

Prefix	Names	Usage
xsd	http://www.w3.org/2001/XMLSchema	XML Schema namespace

Document Attributes

The first section of the document contains the definition of the attributes related to the processing of the document content.

```
<SOAP-ENV:Body>
  <m:submitDocument>
    <Document>
      <Attributes>
        <Attribute name="name" />
      </Attributes>
    </Document>
  </m:submitDocument>
</SOAP-ENV:Body>
```

Here is the list of attributes and their descriptions:

- *csvdelimiter*: String used to separate individual values.
- *csvheader*: Flag indicating if a header line should be shown.
- *csvquote*: String to use to quote individual values.
- *largegraph*: Enables the flag indicating if objects must be loaded type by type instead of one big chunk. For huge loading graphs, it is preferable the load them type by type.
- *locale*: Default locale to use.
- *mode*: The export mode.
- *non.updatable.fields*: Directive to the integration operation on what to do when trying to update a non-updatable field.
- *pagingsize*: Number of records by page.
- *preparator*: Used to specified a service that will be called before the first record and after the last record.
- *switch.system.maintenance*: Switch the zone to maintenance. In some products, it is required to put the zone in maintenance to execute an operation. Required services are documented in the product's data dictionary (e.g.: the JobField entity of the Smartorg product in version 11a).
- *unknown.custom.fields*: Behavior when a UDF name is specified in an import request and not find in the zone.
- *version*: Data mapping version.

Table 5: csvdelimiter

String used to separate individual values.

Name	Value
Service Type	Export
Possible Values	N/A
Default Value	The comma character

Table 6: csvheader

Flag indicating if a header line should be shown.

Name	Value
Service Type	Export

Name	Value
Possible Values	true, false
Default Value	false

Table 7: csvquote

String to use to quote individual values.

Name	Value
Service Type	Export
Possible Values	N/A
Default Value	The double quote character

Table 8: largegraph

Enables the flag indicating if objects must be loaded type by type instead of one big chunk. For huge loading graphs, it is preferable the load them type by type.

Name	Value
Service Type	Export
Possible Values	true, false
Default Value	false (new Query in TCC is default to true)

Table 9: locale

Default locale to use.

Name	Value
Service Type	Import/Export
Possible Values	Language abbreviation
Default Value	There is no default value for this attribute.
Example	en

Table 10: mode

The export mode.

Name	Value
Service Type	Export
Possible Values	T-XML, CSV-ENTITY, CSV, XML
Default Value	There is no default value for this attribute.

Table 11: non.updatable.fields

Directive to the integration operation on what to do when trying to update a non-updatable field.

Name	Value
Service Type	Import
Possible Values	error, warning, error.different
Default Value	error

Value	Equal XML and Entity	Different XML and Entity
error	When trying to update a non-updatable field, returns an error message. Fails the import.	When trying to update a non-updatable field, returns an error message. Fails the import.
warning	When trying to update a non-updatable field, returns a warning message saying that both values have the same value. Does not fail the import but the entity value is not updated with the provided one.	When trying to update a non-updatable field, returns a warning message saying that both values do not have the same value. Does not fail the import but the entity value is not updated with the provided one.
error.different	When trying to update a non-updatable field, returns a warning message saying that both values have the same value. Does not fail the import but the entity value is not updated with the provided one.	When trying to update a non-updatable field, returns a warning message saying that both values do not have the same value. Fails the import.

Table 12: pagingsize

Number of records by page.

Name	Value
Service Type	Export
Possible Values	-1 (no paging), or a positive number
Default Value	-1



The *pagingsize* attribute is not supported when the *mode* attribute has the value CSV.

Table 13: preparator

Used to specified a service that will be called before the first record and after the last record.

Name	Value
Service Type	Import

Name	Value
Possible Values	See data dictionary.
Default Value	None.
Example	olf

Table 14: switch.system.maintenance

Switch the zone to maintenance. In some products, it is required to put the zone is maintenance to execute an operation. Service that required that are documented in the product data dictionary (e.g.: the JobField entity of the Smartorg product in version 11a).

Name	Value
Service Type	Import
Possible Values	always
Default Value	None.

Table 15: unknown.custom.fields

Behavior when a UDF name is specified in an import request and not found in the zone.

Name	Value
Service Type	Import
Possible Values	warn, error, ignore
Default Value	error

Table 16: version

Data mapping version.

Name	Value
Service Type	Import, Export
Possible Values	See data dictionary in the Data Model, Version section.
Default Value	None.
Example	http://www.taleo.com/ws/tee800/2009/01

Document Elements

Table 17: import/export entities

Element	Description
<ImportEntities>	Root element containing a list of operations to perform on an entity.

Element	Description
<ExportEntities>	Root element containing a selection query.

ImportEntities

The element name under ImportEntities has the following form: *Entity-operation*

Entity: The *entity* value corresponds to the entity type name on which the import will be done. The list of possible entities is documented in the product's data dictionary under the **Service > Taleo Connect Client Services > Service Summary** section. Here are some examples from the “Taleo Enterprise, Recruiting Data Dictionary”:

- Candidate
- Department
- Offer
- JobTemplate

An entity always starts with a capital letter and each word starts with a capital letter as well.

Operation: The *operation* value is a bit tricky to construct. The following rules must be applied:

1. The possible operations are listed under the **Service > Taleo Connect Client Services > Service Details** section of the product's data dictionary.
2. The operation name from the data dictionary must be transformed into a T-XML operation name:
 - a. The Documented Operation is composed of one or many word(s). The first word starts with a lowercase letter and each other words starts with an uppercase letter (studlyCaps).
 - b. In the XML Operation, each word must start with a lowercase and must be separated by a hyphen character (-), aka snake-case.

Here are some examples:

Documented Operation	XML Operation
addComments	add-comments
attachFile	attach-file
attachRecruiterFileToApplication	attach-recruiter-file-to-application
bypass	bypass
create	create
match	match
merge	merge
mergeWithEmailControl	merge-with-email-control

