Taleo Integration

Taleo Connect Client
User Guide

Feature Pack 12A
January, 2012
Confidential Information and Notices

Confidential Information
The recipient of this document (hereafter referred to as "the recipient") agrees that the Confidential Information disclosed herein by Taleo shall be retained in confidence by the recipient, and its respective employees, affiliates and/or subsidiaries.

For the purpose herein, the term "Confidential Information" shall mean the following:

1. Any information, know-how, data, process, technique, design, drawing, program, formula or test data, work in process, business plan, sales, suppliers, customer, employee, investor or business information contained in a document, whether in written, graphic, or electronic form; or
2. Any document, diagram, or drawing which is either conspicuously marked as "Confidential", known or reasonably known by the other party to be confidential, or is of a proprietary nature, and is learned or disclosed in the course of discussions, demonstrations, or other collaboration undertaken between the parties.

Limited Rights Notice (Dec 2007)

1. These data are submitted with limited rights under Subcontract No. 6896589. These data may be reproduced and used by the Government with the express limitation that they will not, without written permission of the Contractor, be used for purposes of manufacture nor disclosed outside the Government; except that the Government may disclose these data outside the Government for the following purposes, if any; provided that the Government makes such disclosure subject to prohibition against further use and disclosure: None.
2. This notice shall be marked on any reproduction of these data, in whole or in part.

© 2012 Taleo Corporation. Do not reproduce without the written permission of Taleo Corporation.
Table of Contents

Confidential Information and Notices.................................................................................................................. ii

Introduction
Taleo Connect.......................................................................................................................................................... 2
Taleo Connect Server........................................................................................................................................... 2
Taleo Connect Client........................................................................................................................................... 2
Taleo Connect Users........................................................................................................................................... 2
Documentation.................................................................................................................................................. 3

Installation
Files and Folders................................................................................................................................................. 5
File Names.......................................................................................................................................................... 5
Folder Structure................................................................................................................................................ 6

In Windows............................................................................................................................................................ 7
Installing TCC in Windows.......................................................................................................................... 7
Testing TCC in Windows..................................................................................................................................... 8
Upgrading TCC in Windows................................................................................................................................... 8

In Unix/Linux...................................................................................................................................................... 10
Installing TCC in Unix/Linux..................................................................................................................... 10
Testing TCC in Unix/Linux.......................................................................................................................... 10
Upgrading TCC in Unix/Linux....................................................................................................................... 11

Description
Taleo Connect Client Basics.............................................................................................................................. 13
Ping Dialog........................................................................................................................................................ 13
Eclipse RCP Framework...................................................................................................................................... 14
TCC Menu Bar................................................................................................................................................. 15
Design Perspective............................................................................................................................................ 17
Outline View...................................................................................................................................................... 18
Advanced Queries

Introduction .................................................................................................................. 135
Inserting Complex Queries .............................................................................................. 136
Examples .......................................................................................................................... 137
Between ............................................................................................................................ 137
Case (Switch by criterion) ................................................................................................. 137
Case (Switch by value) ....................................................................................................... 138
Concatenate ....................................................................................................................... 138
Connect By ........................................................................................................................ 139
Count ................................................................................................................................ 139
dbms_lob.getlength ........................................................................................................ 140
dbms_lob.instr ................................................................................................................ 140
dbms_lob.substr ............................................................................................................. 141
Decode .............................................................................................................................. 141
Distinct ............................................................................................................................. 142
Dual ................................................................................................................................... 142
Exists / Not Exists .......................................................................................................... 143
Greater Than .................................................................................................................... 144
Group By .......................................................................................................................... 144
Having ............................................................................................................................... 145
In ....................................................................................................................................... 146
Inner Join .......................................................................................................................... 146
Intersect ............................................................................................................................ 147
Less Than ......................................................................................................................... 147
Like ................................................................................................................................... 148
Minus ................................................................................................................................ 149
NVL ................................................................................................................................... 149
Or ..................................................................................................................................... 150
Order By ........................................................................................................................... 150
Outer Join .......................................................................................................................... 151
Rownum ............................................................................................................................ 151
SubString .......................................................................................................................... 152
ToChar ............................................................................................................................... 152
Trim ................................................................................................................................... 153
Truncate ............................................................................................................................. 153
Union .................................................................................................................................. 154
Union All .......................................................................................................................... 155
Other Custom Functions ............................................................................................... 156
Custom Function Example ............................................................................................ 156
Oracle Reserved Words ................................................................................................. 157

Security

Data Confidentiality ......................................................................................................... 162
User Authentication .......................................................................................................... 163
Stored Passwords ............................................................................................................. 164
Specifying Integration Password .................................................................................... 165
Override ENCRYPTED_PASSWORD .............................................................................. 166
Encrypting Proxy Password ............................................................................................ 167
HTTP Proxy ....................................................................................................................... 168
Encrypting Proxy Password ............................................................................................ 168
Microsoft NTLM V1 Support .......................................................................................... 170
Data at Rest ....................................................................................................................... 171
TCC Plug-ins Security ..................................................................................................... 172

Appendices

Appendix A ....................................................................................................................... 175
Introduction

- Taleo Connect
Taleo Connect

Taleo Connect Server

The Taleo Connect Server is the server-side portion of the Taleo Connect Platform embedded within the Taleo products. It enables the products to expose their integration offering through a secure gateway using system-to-system type messages. The Taleo Connect Server export feature allows access to the Taleo product’s dictionary and its import feature accepts instructions to invoke service operations. The Taleo Connect Server only accepts messages in the proprietary T-XML format wrapped in a standard SOAP envelope with Taleo extensions (named T-SOAP).

Taleo Connect Client

Taleo Connect Client is the client-side portion of the Taleo Connect Platform that resides in the customer’s environment. It enables customers to build and execute their own integration processes.

Development Environment

The development environment is user-friendly and professional. It allows the design and configuration of simple to complex integration scenarios.

The TCC design and configuration environment is built with the powerful and extensible Rich Client Platform (RCP) from the Eclipse foundation. RCP is the standard for Java client user interface development offering a professional grade experience to the end user.

All the user tasks are either wizard driven or based on natural client application behavior such as drop-down menus and drag-and-drop functionality.

Integration Engine

The integration engine is powerful and flexible. It can communicate with the Taleo environment using multiple file formats.

TCC communicates with Taleo products using proprietary Taleo message formats that are based on industry standards such as XML, SOAP, and WS-Addressing. It also supports custom message formats through client-side or server-side transformations since customer systems usually only support their own formats.

TCC shields the complexity of the asynchronous integration process from the customer and provides a synchronous-like behavior that can integrate into other business processes.

Taleo Connect Users

Taleo Integration Users must be created with a System Integrator role. Refer to your SmartOrg Administrator Manual for user creation details.

Taleo recommends the creation of different users for production and staging zones. Use the staging zone user account for integration development and testing. Use the production zone account to go live in production with your integrations.
Documentation

The Taleo Connect Client (TCC) documentation is available on the Content Tab of the Customer Support Portal.

User Guide

This document offers a high-level overview of TCC and an in-depth look at the functional concepts involved in creating and executing integration processes. It is addressed to anyone interested in working with TCC and should be considered mandatory reading before installing or using the product.

Data Dictionaries

A data dictionary is the complete reference for the data model and services of a given Taleo Product. The data model consists of entities with fields and relations between other entities. Entities represent the information stored in the application. The services expose callable methods that allow you to access the data model entities from a client application.

Release Notes

The Release Notes documents contain a description of the new features that are offered for that release.
• Files and Folders............................................................................................................... 5
• In Windows....................................................................................................................... 7
• In Unix/Linux.................................................................................................................... 10
Files and Folders

File Names
Distinctive file names facilitate the management of various integration points.

Touchpoints
- **Import**
  An import touchpoint corresponds to an available Taleo product service offering as defined in the data dictionary. The services have the same name as the business entity for which they offer specific import operations. For example, in the SmartOrg Data dictionary, the Group service offers create, delete, merge, and update operations. Other services may offer different operations.
- **Export**
  An export touchpoint corresponds to the information to be extracted from a particular Taleo product business model.

Actions
The import actions correspond to the specific operations offered by the service. There is only one export action, and that is an export.

File Name Standards
- **Import**
  Load file: `<touchpoint>_<action>_ld.xml`
  Configuration file: `<touchpoint>_<action>cfg.xml`
  Data file: `<touchpoint>_<action>.csv`
  Result file: `<touchpoint>_<action>_result_<date_time>.csv`
- **Export**
  Query file: `<touchpoint>_<action>_sq.xml`
  Configuration file: `<touchpoint>_<action>_cfg.xml`
  Result file: `<touchpoint>_<action>_result_<date_time>.csv`

Examples
- **Import**
  Load file: `user_merge_ld.xml`
  Configuration file: `user_merge_cfg.xml`
  Data file: `user_merge.csv`
  Result file: `user_merge_result_20080715T084011.csv`
- **Export**
  Query file: `newhire_export_sq.xml`
  Configuration file: `newhire_export_cfg.xml`
Result file: newhire_export_result_20080715T090517.csv

Folder Structure

A well organized folder structure facilitates the management, development, testing, and migration from staging to production of the various integration point files.

Level 0
This is the root folder for Taleo integrations. Located here are:

- Zones
  A folder for each zone that is available, production, staging, development, etc...

Level 1
This is the Zones folder level. Located here are:

- touchpoint_action
  A folder for each import and export integration point.

Level 2
This is the integration point folder level. Located here are:

- For import integration points:
  The inbound folder for data files to import.
  The outbound folder for result files.
- For export integration points:
  The outbound folder for result files.

Level 3
This is the import integration inbound folder level. Located here is:

- archives
  The archives folder to save imported data files.
In Windows

Installing TCC in Windows

Installing TCC in Windows enables users to create, test, and troubleshoot; import, export, and configuration specification files; in a graphic user interface environment.

Prerequisite

- One of the following operating systems pre-installed:
  - Windows XP
  - Windows Server 2008 32/64
  - Windows 7 32/64
- Hardware:
  - Design and development workstation, ~1GHz, 256 MB RAM,
  - Large volume production workstation, ~1 GHz, 512MB RAM.
- Software:
  - JRE 1.5.0 or JRE 5.0
  - Product Integration Pack (PIP) setup program
  - Taleo Connect Client setup program
- Installation password supplied by your Taleo Customer Support Team.
- Security approval from your information security department to install TCC.

Integration offer definition file renamed Product Integration Pack. Taleo product release rebranding forced the renaming of feature packs (fp file type) to product integration packs (pip file type).

Integration offer definition files (fp and pip) removed from Taleo Connect Client installation package. Prior to TCC FP11B the installation package included several versions of the products integration offer definition files. As of 11B, the customers must download and install the integration offer definition files they require separately.

Steps

1. Create an [INSTALL FOLDER] in which to install the PIP and TCC folders.
2. Double click the Product Integration Pack setup program.
   The Welcome to the Taleo Connect Client Product Integration Pack Setup Wizard window opens.
3. Follow the installation instructions.
   The ProductIntegrationPack folder containing the pip and fp type files is created in the [INSTALL FOLDER].
4. Double click the Taleo Connect Client setup program.
   The Welcome to the Taleo Connect Client Setup Wizard window opens.
5. Follow the installation instructions.
   In the Completing the Taleo Connect Client Setup Wizard window, Deselect the Launch Taleo Connect Client and View README file check boxes before you click Finish.
   Taleo Connect Client is installed in the [INSTALL FOLDER].
Testing TCC in Windows

Testing TCC in Windows validates the endpoints, the user names and passwords, and communication between TCC and the Taleo products.

Prerequisite

- TCC is installed in Windows.
- Production and staging zones are up and running.
- Integrator User name and password for production and staging zones.

On successful completion of the test the TCC graphic user interface opens. On test failure a Ping Error is displayed. To troubleshoot, refer to the TCC Errors list in Appendix B.

Steps

1. Launch Taleo Connect Client.
   The No product integration pack file configured window opens
   The Ping the Taleo Product window opens.
3. Select the product to connect to from the pull down menu.
   The product to ping is selected.
4. Configure the connectivity information.
   a) Select protocol from pull down menu.
   b) Enter host name.
   c) Enter port number.
   Production or staging zone host name, for example acme.taleo.net or stgacme.taleo.net. Taleo zones uses HTTPS protocol with port 443.
5. Configure the authentication information.
   a) Select authentication.
   b) Enter user name.
   c) Enter password.
6. Click Ping.
   The Ping the Taleo product window closes. An information window indicating the creation of a new endpoint opens.
7. Click OK.

Result

The TCC graphic user interface opens or a Ping Error message appears.

Upgrading TCC in Windows

Upgrading to the latest version of TCC allows users to take advantage of the latest fixes, enhancements, and new features. Taleo recommends following the upgrade process described below to ensure smooth transition and minimal impact on existing integration scripts.

Prerequisite

An [INSTALL FOLDER] for the new version of TCC, different from the current installation location.
**Steps**

1. Download and install the latest TCC version.
2. Download and install the latest product integration packs
   
   To install a product integration pack, unzip the downloaded file into the [INSTALL DIRECTORY]/productintegrationpacks directory.
3. Create the configuration board, in order to run existing integration scripts that use configuration board symbols or endpoints.
4. Synchronize the custom dictionaries in order to have custom fields.
5. Set the last run date to the actual value in export configurations that use a LRD pre-processing step.
6. Test the existing integration scripts with the new TCC version.
   
   If using NetChange, TCC users MUST run the “NetChange Setup” to create the NetChange database for the new TCC version.
7. Modify any existing reference to the TCC installation location to point to the new location.
**In Unix/Linux**

**Installing TCC in Unix/Linux**

Installing TCC in Unix/Linux enables users to run import and export configuration files, in a command line environment.

**Prerequisite**
- Operating system:
  - Unix
  - Linux
- Hardware:
  - Large volume production workstation, ~1 GHz, 512MB RAM
- Software:
  - JRE 1.5.0 or JRE 5.0
  - TaleoConnectClient-X.Y.Z.tar.gz file
- Security approval from your information security department to install TCC.

**Steps**
1. Copy the TaleoConnectClient-[version].tar.gz file to your UNIX or Linux environment.
2. Unzip the file with this instruction, gunzip TaleoConnectClient-[version].tar.gz
   
   The following file is generated, TaleoConnectClient-[version].tar.
3. Untar the file with this instruction, tar –xf TaleoConnectClient-[version].tar.

**Result**

The Taleo Connect Client is installed in the tcc-[version] directory.

**Testing TCC in Unix/Linux**

Testing TCC in Unix/Linux validates the conformity of the integration workflow between TCC and the Taleo products.

**Prerequisite**
- TCC is installed in Unix/Linux.
- Production and staging zones are up and running.
- Unix/Linux system has access to zones.
- Valid import specification and data file, and associated configuration file.
- Valid export specification and associated configuration file.

**Steps**
1. Invoke import configuration execution.
2. Invoke export configuration execution.
3. Verify monitoring files.
4. Verify result files.

Upgrading TCC in Unix/Linux

Upgrading to the latest version of TCC allows users to take advantage of the latest fixes, enhancements, and new features. Taleo recommends following the upgrade process described below to ensure smooth transition and minimal impact on existing integration scripts.

Prerequisite

An [INSTALL FOLDER] for the new version of TCC, different from the current installation location.

Steps

1. Download and install the latest TCC version.
2. Download and install the latest feature packs
   To install a feature pack, unzip the downloaded file into the [INSTALL DIRECTORY]/featurepacks directory.
3. Copy the configuration board from [OLD INSTALL DIRECTORY]/system/default.configuration_brd.xml to [NEW INSTALL DIRECTORY]/system directory, in order to run existing integration scripts that use configuration board symbols or endpoints.
4. Copy the last run date files from [OLD INSTALL DIRECTORY]/lastrundates/*.xml to [NEW INSTALL DIRECTORY]/lastrundates directory.
5. Test the existing integration scripts with the new TCC version.
   If using NetChange, TCC users MUST run the “NetChange Setup” to create the NetChange database for the new TCC version.
6. Modify any existing reference to the TCC installation location to point to the new location.
• Taleo Connect Client Basics......................................................................................... 13
• Default Configuration Board........................................................................................ 24
• Import............................................................................................................................. 29
• Export........................................................................................................................... 40
• Configuration............................................................................................................... 52
• Runtime....................................................................................................................... 78
• Console......................................................................................................................... 80
• Command Line............................................................................................................. 85
Taleo Connect Client Basics

Ping Dialog

The ping dialog window (Ping the Taleo product) opens when Taleo Connect Client is started. The ping dialog validates the endpoints, the user names and passwords, and communication between TCC and the Taleo products.

The ping dialog uses the default configuration board endpoint settings. The original default board file is provided by your Integration Customer Support representative. If this file has not been provided, you can manually enter zone information in the Endpoint and the Authentication sections.

### Connectivity Information
- **Product:**
  - The product to ping is selected from the pull down menu.
- **Company Code:**
  - The company code is used by Taleo Business Edition users.
- **Endpoint:**

### Ping the Taleo product

#### Connectivity information

Provide the connectivity information to access the product:

- **Product:** Professional 7.5.5
- **Company Code:**

#### Endpoint

Configure the connectivity information:

- **Protocol:** HTTPS
- **Host:** acme.taleo.net
- **Port:** 443

#### Authentication

Configure the authentication information:

- **No authentication**
- **Basic authentication**
- **User Name Token authentication**

- **User:** johndoe
- **Password:** *********

[Import Endpoint] [Ping]

Connectivity Information

- **Product:**
  - The product to ping is selected from the pull down menu.
- **Company Code:**
  - The company code is used by Taleo Business Edition users.
- **Endpoint:**
The Endpoint to use is selected from the pull down menu.

**Endpoint**
The Endpoint section displays the settings of the selected Endpoint. If no endpoints are configured, the information can be entered manually.

- **Protocol:**
  The protocol to use is selected from the pull down menu.

- **Host:**
  Production or staging zone name, for example acme.taleo.net or stgacme.taleo.net.

- **Port:**
  The port number to access the Taleo product. Port 443 is used by default.

**Authentication**
No authentication is listed as an option for testing purposes. User/password HTTP Basic Authentication is the minimal requirement to connect to the Taleo environment. User Name Token authentication offers improved security by encrypting the password in all files.

- **User:**
  The user name for integration must be created with an integrator role.

- **Password:**
  The integration user password.

**Import Endpoint**
The Import Endpoint link opens a browser window that allows the user to locate the original default configuration file supplied by Customer Support or a legacy environment properties file and retrieve the endpoint settings.

**Ping**
The ping button starts the process. On success TCC starts, on failure an error message is displayed. Refer to the error messages in Appendix B for additional details.

**Eclipse RCP Framework**
The Taleo Connect Client integration development environment is based on the Eclipse foundation's Rich Client Platform project (http://www.eclipse.org). It is important to understand the underlying concepts of this framework before going through the features in Taleo Connect Client.

**Underlying Concepts**
A **perspective** is a logical grouping of elements that organizes the work for the user in a clear and uncluttered fashion (including views, tool bars, and menu items).

An **editor** is a user interface that allows a user to create and update a particular part of the integration process. Multiple editor types and instances may be opened in a single perspective.

A **view** is a read-only auxiliary element that offers information about the active editor. As such, views react to selections within the active editor and to changes to different editor instances.

A **preference** is a setting that affects the user experience within the application. All preferences have a default value set by Taleo that can be changed by the user and preserved between work sessions.
TCC Menu Bar

The TCC menu bar gives access to file utilities, a report viewer, window utilities, and the help system.

File

- **New**
  - New Configuration Wizard: Opens a blank configuration specification in the editor.
  - New Import Wizard: Opens a blank import specification in the editor.
  - New Export Wizard: Opens a blank export specification in the editor.
  - New Profile Wizard: Opens a blank console profile in the editor.
- **Open**
  - Configuration: Opens a browser to select an existing configuration specification to open in the editor.
  - Import: Opens a browser to select an existing import specification to open in the editor.
  - Export: Opens a browser to select an existing export specification to open in the editor.
  - Console Profile: Opens a browser to select an existing console profile to open in the editor.
  - Configuration Board: Opens the Default Configuration Board in the editor.
- **Close**
  - Closes the active file in the editor.
- **Close All**
  - Closes all files in all editors.
- **Save**
  - Saves the active file with the same name in the same folder.
- **Save As...**
  - Opens a browser window to save the active file in a specific folder as a specific file name.
- **Exit**
  - Closes Taleo Connect Client

Report

- **Schedule monitoring...**
  - Opens the Schedule Monitoring Report. The report lists the Scheduled processes, the next expected run date, and the last run status and date.

Window

- **Open Perspective**
  - Console
  - Design
  - Runtime
- **Show View**
  - Basic
  - Client
  - Document
  - General
• Help
  • Message
• Preferences
  • Client>Design
    Sample CSV viewer: CSV file editor that opens during sample CSV file creation. Microsoft Excel is the editor by default.
    Maximum depth of the entity tree: Advanced setting, do not change value unless directed by Taleo Customer Support.
  • Client>Runtime
    Monitoring auto-refresh rate (seconds): Advanced setting, do not change value unless directed by Taleo Customer Support.
    Number of files to keep in monitoring history: Advanced setting, do not change value unless directed by Taleo Customer Support.
• Console
  Advanced settings, do not change values unless directed by Taleo Customer Support.
• General
  Advanced settings, do not change values unless directed by Taleo Customer Support.
• Help
  Advanced settings, do not change values unless directed by Taleo Customer Support.
  • Restore Defaults
    Restores all setting to their default values.
  • Apply
    Applies new settings to Taleo Connect Client.

Help
• Help Contents
  Opens the online help window.
• About Taleo Connect Client
  • Feature Details: Opens a list of TCC features.
  • Plug-in Details: Opens a list of installed plug-ins.
  • Configuration details: Displays the TCC configuration details.
Design Perspective

The design perspective is where the integration process workflow is built and where you determine how to extract information from or upload information to the Taleo environment. A specific editor is offered for each of these activities.
Outline View

The Outline view shows a structured view of the content of the current editor.
Entity View

The Entity View shows the structure of the entity selected within an Export Editor or a service operation parameter selected within an Import Editor.

Each service operation offers a specific set of parameters that can be used to define the characteristics of data columns. Refer to the data dictionary for additional details.

The root entity has a set of fields and relations. The information section at the bottom of the view displays the details of the selected element.
Product Integration Pack View

The Product Integration Pack View shows the entire integration offering of a Taleo product with the available standard content.

Model

The model section offers a selection of the installed product integration packs. Each product has typically a single model, although previous models may be included for backwards compatibility purposes.

The state of the dictionary is indicated for the displayed product model. The Synchronize custom fields button is used to update the data dictionary with user defined fields found in the Taleo product.

Dictionary tab

The dictionary tab describes the entities available in the business model. These entities can be searched with an update-as-you-type filter box or with available for export or import filter buttons.

Each entity has a set of fields and relations. The relationships can be navigated to the related entity by double clicking, and back to the original entity with the return button.
Services tab

The services tab lists the services available for this product. The services can be searched with an update-as-you-type filter box. The available operations are specific to each service.
Templates tab

An import template is used to create an import specification for the given entity. The columns provided in the template are typical of integration processes with the Taleo product.

An export template is used to create an export specification for the given entity. The projections (fields) provided in the template are typical of integration processes with the Taleo product.

An export filter is used as a specific filtering element in an export specification. Export filtering elements from the Product Integration Pack are resolved only at runtime in the Prepare Export step; that is, the source export file does not contain the definition of the filtering, but only a reference to the filtering name.
Log View

The Log View lists all error messages produced by the various editors and views.

- The first column shows the severity of the message with an icon.
- The Date column displays the date of the error.
- The Source column displays the source of the error.
- The Message column displays the error message.

![Log View Example](image)

Diagnostic View

The Diagnostic View shows the warnings or errors associated to the current Editor.

- The first column displays the type of Editor that generated the diagnostic message with an icon.
- The second column displays the severity of the message with an icon.
- The File column displays the file name of the Editor that generated the message.
- The Message column displays the actual description of the diagnostic message.
- The Detail column displays additional details.

Jobs View

The Jobs View lists the jobs running in the background.

Documentation View

The Documentation view displays field and relation descriptions from the data dictionary, for the element selected in the Entity or Production Integration Pack views.
Default Configuration Board

Board Editor

The board editor enables users to create endpoint and symbol elements and save them in the default.configuration_brd.xml file for later use.

The Default Configuration Board facilitates migration from one environment to another (Staging to Production, Windows to Unix/Linux) by centralizing endpoint and symbol information in one file.

The default configuration board file location is \TCC_HOME\system\default.configuration_brd.xml. This path and file must be copied to the production environment (Windows, Unix, Linux) to use centralized symbols and endpoints.
Configuration Board Elements

An ordered list of all the configuration board elements is displayed. The remove button deletes selected board elements. The Add button creates new board elements.

Endpoint Element

- **General**
  
  The reference name is what is displayed in the ordered list and in the pull down menus when endpoint information is required.
  
  The Default check box assigns the selected endpoint to be used by default.

- **Endpoint**
  
  The Endpoint section provides the actual connectivity information for the Taleo environment:

  - Protocol: Taleo zones use HTTPS
  - Host: Production or staging zone name, for example acme.taleo.net or stgacme.taleo.net.
  - Port: Taleo zones use port 443.

  The communication path can be set by default, be provided by the full Path directly, or by a combination of the service dispatcher Path and Service Name. The complete web service path can also be provided.

- **Authentication**
  
  The Authentication section provides the authentication context for the requests.

  No authentication is listed as an option for testing purposes.

  User/password HTTP Basic Authentication is the minimal requirement to connect to the Taleo environment.

  User Name Token authentication offers improved security by encrypting the password in all files.

Symbol Element

Symbols can be used anywhere in a configuration file and their values are resolved when executing the integration job. These are name and value pairs.

For example, Name=OUTPUT_FOLDER and Value=c:\tcc\output. The symbol is resolved to the value set in the default configuration board when it is used in a configuration file.
Note: Windows supports backslash (\) and forward slash (/) as path delimiters. When production and staging zones run under UNIX or Linux operating systems, it is recommended to use forward slashes in configuration files since this is the only path delimiter supported by those systems. This simplifies moving integration configurations from staging to production environments since no modifications are required to the file paths in the configuration files.

- **NOW ("DateTimePattern")**

  The symbol will be replaced by the current date and time. The date and time is resolved only once for a configuration execution. Therefore, if you refer to the NOW( ) symbol in a preprocessing step, it will be that resolved value.

  The default date format is: NOW("yyyyMMdd'T'HHmmss") = 20080702T10:12:00

  You can specify the time format by using any supported date and time pattern: NOW("EEE, MMM d, yyyy") = Wed, Jul 2, 2008

  For more information on supported date and time patterns, please refer to: [http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html](http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html)

- **NUMBER_FORMAT ([symbolName])("pattern")**

  The symbol will be replaced by the formatted number. This token has two parameters which are the number to format and the pattern to apply on the number to format.

  The pattern parameter can be composed of the following characters:
  - 0 - Digit
  - # - Digit, zero shows as absent
  - . - Decimal separator or monetary decimal
  - , - Grouping separator

  For example, [NUMBER_FORMAT([itk_attribute_count])("0000000000000")] or [NUMBER_FORMAT([itk_attribute_count])("#,##0.00")]

- **CFG_FOLDER**

  The symbol will be replaced by the configuration file folder location.

  The configuration file location is: c:\tcc\myjob\candidateexport_cfg.xml.

  The token will be replaced at runtime by: c:\tcc\myjob

- **UUID**

  The symbol will be replaced by a universally unique identifier. This is the same token that can be used in the workflow identifier.

- **HOST**

  The symbol will be replaced by the host name of the workstation. This is the same token that can be used in the workflow identifier.

- **RANDOM**

  The symbol will be replaced by a random integer. This has the same purpose as the UUID but it is smaller (32 bits). There is no warranty to be globally unique.

- **SYSTEMPROPERTY("system.property.name")**

  The symbol will be replaced by the specified property name.

  For example: [SYSTEMPROPERTY("java.io.tmpdir")]\tcc, places temporary files in the tcc folder under the system's default temporary directory.

- **Overriding symbol value**

  When executing TCC via the command line, you can override any symbol by specifying a system property with the following pattern: `com.taleo.client.symbol.TOKEN_NAME`
For example, to override the [OUTPUT_FOLDER] symbol defined in the default configuration board, specify the following system property when launching TCC:

-Dcom.taleo.client.symbol.OUTPUT_FOLDER=c:\folder\override

Creating Endpoints

Prerequisite
Taleo Connect Client is running.

Steps
1. File>Open>Configuration Board
   The board editor opens.
2. Click Add.
   A popup dialog opens.
3. Select Add a new endpoint element.
4. Click OK.
   A new Endpoint board element appears in the list
5. Enter the endpoint information.
   a) Enter reference name.
   b) Select default to set this endpoint as the default endpoint.
   c) Select protocol from pull down menu.
   d) Enter host name or IP address.
   e) Enter port number. *Taleo products use port 443.*
   f) Select service dispatcher path, and enter information if necessary.
   g) Select authentication method.
   h) Enter user name.
   i) Enter password.
6. Click Save.
7. Close board editor.

Creating Symbols

Prerequisite
Taleo Connect Client is running.

Steps
1. File>Open>Configuration Board
   The board editor opens.
2. Click Add.
   A popup dialog opens.
3. Select Add a new symbol element.
4. Click OK.
A new symbol board element appears in the list.
5. Enter symbol name.
6. Enter symbol value.
7. Click Save.
8. Close board editor.
Import

Import Editor

The Import Editor creates a specification to upload comma separated value (CSV) files into a Taleo product.

Each service operation supports a list of parameters that can be simple values or dictionary entities. Each column is associated to a particular parameter. For entity type parameters, each column is associated to a field or relation of the entity itself.

General Tab

The general tab presents information that applies to the whole load file.

```
General

Operation information
Describes the selected import operation
Product  : Professional 7.5.6
Model    : Professional 7.5
Service  : Candidate
Operation: merge

Locale
Select the default locale
Locale   : English

Other information
Configure the other import information
☑️ Header present
Value delimiter : ,
Quotation character : "
Advanced : Other import options...
```

Operation information

The Operation information describes what import service operation of the Taleo product offering is used. This information is set when creating the import specification file and cannot be changed.

Listed are:

- Product
• Model
• Service
• Operation

Locale
The Locale determines the language used when uploading multilingual values for columns where the default locale strategy has been selected.

The default language for multilingual fields, such as descriptions, is selected from the pull down list.

Other information
The CSV format determines the characteristics of the request message.

When **Header present** is checked, it indicates that the source file contains an initial record containing the header of the columns. It is required to prevent using the values of the header line as an actual upload record.

**Value delimiter** determines the character used to separate values within a particular record. CSV file values can be separated by commas (,), pipes (|), or other characters. Taleo recommends the comma as a value delimiter for CSV files.

**Quotation character** determines the character used to quote (i.e. surround) a particular value that may contain special characters, such as the value delimiter. Taleo recommends the double quote (") as quotation character.

Fields with embedded double-quote characters must be delimited with double-quote characters, and the embedded double-quote characters must be represented by a pair of double-quote characters. For example, width 50":

• true,"width 50" etc",abc is valid
• true,"width 50" etc",abc is invalid

Advanced: Other import options
The integration processor executes specific code in the Taleo environment before the first imported entity and after the last imported entity. The mandatory integration processor is automatically added when selecting the desired import service. For OLF entity imports, the "olf" integration processor is added. For Department delete, the "maintenance" integration processor is added.

Additional import options are used to toggle specific behavior when processing the integration document. These are name and value pairs.

• **switch.system.maintenance**
  
  This option allows you to automatically switch the zone to maintenance mode when importing Organization, Location, or JobField (OLF), or deleting Department.

  • Possible values:
    
    • **always**: The integration processor will force the zone to switch into maintenance mode.
    • **never**: When importing OLF entities or deleting Department entities, the result file displays “To make any changes, the system must be in maintenance mode.”.

• **non.updatable.fields**

  Special behavior when trying to update a non-updatable field. This is very useful when calling a merge service where some values are required to create a new entity but cannot be updated.

  • Possible values:
    
    • **error**: the record update will fail when non-updatable fields are found during the import. This is the default behavior.
    • **error.different**: the record update will fail when trying to change the value of a non-updatable field.
• **warn**: will never try to update the field value. A warning will be logged within the integration message saying the value has not been updated.

• **info.import.limit**
  
  This option allows you to retrieve information about your daily import limit and count.
  
  • **Possible values:**
    
    • **true**: the response SOAP document contains the `info.import.daily.limit` and the `info.import.daily.count` attributes.
    
    • **false**: the limit and count attributes are absent from the response SOAP document.

### Columns Tab

A Column defines the conversion rules for a particular value of a record of the source file.
**Column**

An ordered list of all the columns in the import specification is built by dragging and dropping fields from Entity view Structure section. The Up and Down buttons allows sorting of the fields in the required order. The remove button deletes selected column. The Add button creates new columns:

- **data column**
  Field from Entity view Structure section. The different service operations offer specific parameters that can be used as data columns.

- **locale column**
  Special column indicating locale, it is used with Multi-lingual fields Record locale type.

- **identifier column**
  Adding an Identifier column simplifies troubleshooting the error messages in the results file. It is not processed by the Taleo product.
  Use the same value as the key field for the record. It will be returned in the result file with a mention of success, error, or failure.

- **ignore column**
  An ignore column is never processed. It is used when the customer system generates a column that has no equivalent in the Taleo product.

There are several import field types:

- Search or lookup fields are identified with a magnifying glass icon.
- Key fields are identified with an orange key icon (visible after checking Use the value for search or lookup).
- Non updatable fields are identified with a red circle icon.
- User defined fields (UDFs) are identified with a person icon.

Numerous column options can be set in the following sections. Some options are field specific. To set these options, click on the column to select it.

**Column information**

The header is a label used to identify the column for information purposes only. The value is disregarded during the actual integration process.

The Field entry determines the actual field of the root entity. Values can be dragged here from the Entity Structure view.

**Behavior**

The Behavior section determines what kind of information the value provides to the import.

**Use the value to search or lookup**: The column may be used to search within the Taleo product for existing entities. There are two types of searches:

- **Key column search**: The search value affects (targets) the root entity. A key icon marks the column. As such, a key is used to actually search for the entity within the Taleo product. A record may have several keys to refine the search; however, an update can only be executed on one entity at a time. As such, the search keys must turn up either no entities or exactly one; any other result will cause an error.

- **Lookup column search**: The search value does not affect (target) the root entity. A lookup icon marks the column. It is used to create a relation between the entity and another entity within the Taleo product. A typical example of lookups are locations. For example, when creating a requisition for Canada, we need to link the entity to the Canada location, not create a new Canada entity. In particular, a field of the root entity can never be looked up; a relation is required for this particular type of search.
- **Target**: When importing a field that is a relation to another entity, not a standalone field, it will use the value to lookup the Target and bind it to the main entity, rather than update it. For example, if you want to use the Loginname as the search key for your User import, click the browse button and select User.

- **Also set this value in the entity**: When checked, it means that the import will try to update the field. It will not work for a lookup, but will work for a search. For example, if you try to import candidates and you use the email address as a search key, it will also set the email address if it creates a new candidate.

**Start a new element in the collection**: Some relationships are collections, because an entity can have multiple occurrences for that relationship. In the Basic Question import template, multiple answer questions mandate the use of more than one answer. To support this, each of the Answer columns (Question, PossibleAnswers, Description field) is marked as starting a new PossibleAnswer element of the PossibleAnswers collection.

- **Target**: For example, to import 2 job-specific applications for the same candidate with at least 1 work experience for each application. Change the Target of the collection for the 1st and 2nd requisition to PreselectionApplication, but leave the Target of the option Use the value to search or lookup as is.

**Start a new parameter collection item**: The behavior of this option is the same as Start a new element in the collection, but for parameters rather than entities. For example, if you want to import multiple motives in a candidate move (CSW), you must find the motive in the Parameter list box, and then check this option for each motive occurrence. Note that this option is not frequently used.

**Skip column value if empty**: The column may be ignored if the value provided in the file is empty. This is the case mainly when an entity field is optional but does not support the empty string as a valid value. In the Basic Question import template, not all questions will have the same amount of answers. However, you cannot import an empty answer in a question. To support both these restrictions, all answer related columns are marked as optional. Although the CSV file must always contain the total number of values in each row, when certain are empty the answers are not created.

**Value is the location of a file to attach**: The column value may be the location of a file that is to be attached to an entity. In such cases, the content of the file is compressed using the G-ZIP algorithm and then encoded in the base-64 scheme to ensure data integrity when embedding the file in the T-XML document. For example, the CurriculumVitaeLocation column (AttachedFiles, Content field) is configured to contain the location on the local system of a file to attach to the candidate. The file needs no preparation nor compression nor encryption and all Taleo supported formats are accepted (HTML, text, Microsoft Word, etc).

**Reset the collection**: The default and only possible value for this parameter in TCC is TRUE. To change the default behavior, enter the value FALSE in the corresponding column of the CSV file. The value set in the CSV file will override the default TCC value for that parameter.

When “Reset the collection” is TRUE, the elements of that collection must not check the “Use the value to search or lookup” parameter because those elements are removed and therefore don’t exist.

Please refer to the product data dictionary for the list of resetable relations.

**Multi-lingual fields**

Multi-lingual fields can be imported in several different configurations.

- **Default locale**: Uses the Default locale set in the General Tab for that column.
- **Specific locale**: Uses the locale set in the Locale pull down menu for that column.
- **Record locale**: Uses the Locale column to identify field locale value. When this option is selected for a column, the presence of a single locale column is mandatory. The value of this column should be a standard locale code such as ‘en’ or ‘fr’.
Advanced options

Advanced options are used to toggle specific behavior when processing the column value. These are name and value pairs.

- **default.value**
  
  A default value can be specified for columns that have no value set in the customer CSV file. Processing is not executed when “Skip column if value empty” is checked.

- **updatable**
  
  Some field values only need to be sent to the target application once, usually during the create operation. Once the column is defined as non-updatable, the field values contained in that column are not sent to the target application in an update operation. This advanced option is Net Change specific.

  - Possible values:
    - true: will update the field (default)
    - false: will not update the field

Processing Instructions Tab

Processing instructions allow customization and extension of import processing on fields and relations.

A processing instruction is a name value pair that can be set on any field or relation available for import in TCC. The attribute name is a predefined string understood by a component in TCC indicating that special processing is required. The instruction is processed locally by TCC, before or after the import depending on the particular instruction.
Processing Instructions List
An ordered list of all the fields and relations that require processing instructions is built by dragging and dropping them from the Entity view Structure section. The Up and Down buttons allows sorting the fields and relations in the required order. The remove button deletes selected fields and relations. The Add button creates a blank processing instruction subject.

Processing Instruction

• **Path**
  
  Path of the field or relation for which a processing instruction is specified.
  
  For a blank processing instruction subject, the path to a valid field or relation can be entered manually.

• **Instructions**
  
  The add button enables the user to enter the instruction name value pair. Only predefined instructions are understood by TCC. Setting an invalid or unknown instruction will be ignored and have no effect.
  
  The remove button deletes the selected instruction.

NetChange Partial Load
The NetChange partial load processing instruction limits the extraction of collection type relation items to those specified in the customer data file during NetChange Setup. This subset of relation items is used to populate the NetChange database. This allows customers to work with partial collection data sets without creating deletes in their zone with ongoing NetChange runs.
• `netchange.load.partial`
  Possible values are true (case-insensitive) or false. Anything other than true is considered false.
  Setting this processing instruction on a field rather than on a relation has no effect.

## Creating a New Import File

**Prerequisite**

Make sure the Taleo Connect Client application is open.

**Steps**

1. *File>New>New Import Wizard>Next*
2. Select a service from one of the available product models.
   a) Select product from pull down menu. The Model is automatically selected based on the product.
   b) Select Import Service from list. The available services have the same name as the entity they act on. A filter is available to narrow down the choices.

3. Select file base.
   Create new import specification is selected by default. This will create an empty import specification.
   a) Select Operation from pull down menu.
Templates are available for certain basic services. When one of those services is selected, Use the following import template becomes available. This will create a populated import specification.

a) Select Use the following import template.
b) Select Template from pull down menu.

4. Click Finish.

The New import specification is displayed in the import editor.

Saving a New Import File

**Prerequisite**
Make sure a new import specification file is open in import editor.

**Steps**
1. File>Save
2. Select directory.
3. Enter file name or select existing name in directory.
4. Select file type.
5. Click Save.

Opening an Existing Import File

**Prerequisite**
Make sure the Taleo Connect Client application is open.

**Steps**
1. File>Open>Import
2. Select directory.
3. Select file in directory.
4. Click Open.

Building an Import Specification

**Steps**
1. Verify that the information in the General tab is correct.
2. Click the Column tab.
3. Create columns.
   a) Drag and drop the required fields from the entity structure view to the column view.
   b) Set column behavior.

Creating a CSV File

**Prerequisite**
A configured and saved import specification is open in the import editor.
Steps
1. Click Generate a sample csv file.

   A sample csv file with the import specification columns opens in the default editor.
2. Enter column data.
3. Save as csv file.

Changing Default CSV Editor

Prerequisite
TCC is open.

Steps
1. Open Preferences window Window>Preferences>Client>Design.
2. Select Client>Design.
3. In Sample CSV viewer, click open window button [...].

   A Select file window opens.
4. Navigate to new CSV editor folder.
   a) Select CSV editor.
   b) Click Open

   The path and name of the new default CSV editor are displayed in the Sample CSV viewer dialog box.
5. Click Apply then OK.
Export

Export Editor

The Export Editor uses a business model from a particular Taleo product version to create a specification for information to be extracted from Taleo products.

Each export is based on a root entity from which all fields and relations are defined. From that entity, fields and relations may be selected (projected) for extraction from the Taleo product.

General Tab

The general tab presents information that applies to the whole export file.

![Export Editor screenshot]

**General**

**Entity information**

Describes the selected exportable entity

- **Product**: Professional 7.5.6
- **Model**: Professional 7.5
- **Entity**: Candidate

**Locale**

Select the default locale

- **Locale**: English

**Other information**

Configure the other export information

- **Alias**: 
- **Export mode**: CSV-entity
- **CSV header present**: 
- **CSV value delimiter**: 
- **CSV quotation character**: 
- **CSV record delimiter**: 
- **Advanced**: Other export options...
Entity information

The Entity information describes what entity of the Taleo product offering is exported. This information is set when creating the export specification file and cannot be changed.

Listed are:

- Product
- Model
- Entity

Locale

The Locale determines the default language used when extracting multilingual fields for projections where the default locale strategy has been selected.

The default language for multilingual fields, such as descriptions, is selected from the pull down list.

Other information

The Alias is used to identify the query and reuse return results in sub queries.

The Export Mode determines how the extracted information is returned in the response message:

- **T-XML**: Uses a structured XML format (akin to a tree view) to represent the data. This mode is well suited to complex data sets including several relations, for example, exporting new hires with their applications, the associated requisition (job offer) information, the work experiences, the education elements, etc. The two main limitations of the T-XML format is that only fields of the business model are allowed and that due to the structure format, the performance of large exports is usually better in the other modes.

- **XML**: Uses a flat XML format (akin to a tabular list or an Excel spreadsheet) to represent the data. This mode is well suited for large exports of simple data sets with no or only direct relations. For example, exporting all of the users of a Taleo product or all of the currently opened requisitions. The main limitation of the XML mode is that when n-ary relations are involved, one record per combination of relations is exported (i.e. for a candidate with two work experiences and three education elements, a total of 6 records would be exported).

- **CSV-entity**: Based on the T-XML export mode, it handles multiple values and multilingual fields. All data related to the root entity is located on the same line. When a column contains multiple values (multilingual or relations of maximum cardinality "N"), the values are serialized inside a single column. The prevent duplicates, grouping, and joining features are not supported by the CSV-entity export mode.

- **CSV-report**: Uses a flat file format (that can be directly imported as an Excel spreadsheet) to represent the data. This mode exports exactly the same information as the XML mode, hence has the same strengths and weaknesses.

The following information only applies to the CSV modes.

When **Header present** is checked, it indicates that the projection aliases and field names will be exported as the first record.

**Value delimiter** determines the character used to separate values within a particular record. By default the export uses the comma (,) character.

**Quotation character** determines the character used to quote (i.e. surround) a particular value that may contain special characters, such as the value delimiter. By default the export uses the double quote (" character).

The **Record delimiter** determines the character used to separate records within the file. By default the export uses the new line character and thus places each record on a separate line.
Advanced: Other export options

The Advanced export options should only be used by Taleo certified specialists to tune the performance of certain large extractions.

Additional export options are used to toggle specific behavior when processing the integration document. These are name and value pairs.

• info.export.limit
  This option allows you to retrieve information about your daily export limit and count.
  • Possible values:
    • true: the response SOAP document contains the info.export.daily.limit and the info.export.daily.count attributes.
    • false: the limit and count attributes are absent from the response SOAP document.

Projections Tab

A Projection defines which fields of the entity will be returned.

The use of projections differs greatly depending on the mode of the export specification:

• When T-XML or CSV-entity mode is used, the only possible projections are field projections. The sequence of the projections is irrelevant as they only serve to identify which elements of the business model will be selected for extraction after the entities have been determined using the filtering elements.
• When a flat mode such as XML or CSV-report is used, projections may include transformations. In this mode, the sequence of projections is critical as they each represent a column in the resulting document.
Projections

An ordered list of all the projections in the export specification is built by dragging and dropping fields from Entity view Structure section. The Up and Down buttons allows sorting of the projections in the required order. The remove button deletes selected projections. The Add button creates new projections:

- **field projection**
  
  A field projection is the most common projection and directly represents an element of the business model. The alias of a field projection is optional. When an alias is not present, the path itself will be used. The field is defined by a path that navigates the business model starting from the root entity set for the export specification.

  For multi-lingual field projections, the extracted locale values can be selected in the Locale. Filtering window that is opened with the Field open window button.

- **formatted date projection**
  
  A formatted date projection allows modification of the field date format. The alias of a formatted date projection is mandatory. The date field is defined by a path that navigates the business model
starting from the root entity set for the export specification. The date format defines the presentation of the date elements.

- **function projection**

  Available functions are: Add, Add days, Add weeks, Add months, Add years, Average, Cast as date, Cast as number, Cast as timestamp, Concatenate, Count, Decode, Divide, Greatest, Least, Maximum, Minimum, Multiply, Power, Replace, Replace null, Replace null or not, Round, Substring, Subtract, Subtract date, Sum, To char, To end of day, To start of day, To lower-case, To upper-case, Trim, Truncate

  The alias of a function projection is mandatory. The plus and minus buttons add and delete parameters. The up and down symbols moves the selected parameter. Available function parameter data types are: String, Boolean, Integer, Long, Float, Double, Date, Date/Time, and Field. Parameter values must conform to the selected data type.

- **multilingual field projection**

  Allows the language code of related entity fields to be exported in CSV-report format. The alias of a multilingual field projection is optional. When an alias is not present, the path itself will be used. The value exported can be the language code (2), or locale code (en).

- **complex projection**

  Insert the SQ-XML code into the complex projection pane.
Filters Tab

Filters determine which entities are extracted from the Taleo product.

Filters

List of all the filters in the export

Two parameter operator filter

Specify operator and values

- Value: Birthday
- Data type: Field
- Operator: Between
- Value: 2008-07-24 16:53:19
- Data type: Date/Time

Filters

The use of filters is the same for all export modes. The sequence of filtering elements is unimportant since they are combined by a logical AND to select entities. The Up and Down buttons allows sorting of the filters if so desired. The remove button deletes selected filters. The Add button creates new filters:

- simple filter

  Field filtering selects entities based on the value of the field. The field can be compared in a number of ways to a specific value that should be of the same type as the field itself.

  Available operators are: Between, Equals, Equals (allow null), Does not equal, Greater than, Greater than or equal, Is null, Is not null, Is true, Is false, Less than, Less than or equal, Contains (case sensitive), Contains (case insensitive), Matches (case sensitive), Matches (case insensitive), and Last Run Date.
Last Run Date filters the records that were added or modified since the last export.

- **complex filter**
  Insert the SQ-XML code into the complex filter pane.

- **filter from a template**
  The template pull down menu displays the available templates for the selected entity.

**One/No parameter operator filter**

The one parameter operator filter requires the argument to be specified. The no parameter operator filter argument is embedded in the operator.

The available parameter data types are: String, Boolean, Integer, Long, Float, Double, Date, Date/Time, and Field.

**Sorting Tab**

Sorting defines the order in which the selected entities will be returned from the Taleo product.
**Sorting**

The use of sorting is the same for all export modes. The sequence of the sorting elements is important as the first sorting will be applied first and the subsequent sorting only for matching entities. The Up and Down buttons allows placement of the sorting elements in the required order. The remove button deletes selected projections. The Add button creates new sorting elements:

- **field sorting**
  
  Field sorting is the most common sorting and orders entities based on the value of the field. The only option is to select the Order of the sort, which can either be Ascending or Descending.

- **complex sorting**

  Insert the SQ-XML code into the complex sorting pane.

- **field based projection sorting**

  When exporting in T-XML or CSV-entity, field based projection sorting allows specifying the sorting order of sub-entities. The only option is to select the Order of the sort, which can either be Ascending or Descending.

- **complex projection sorting**

  Insert the SQ-XML code into the complex projection sorting pane.
Advanced Tab

The advanced tab enables users to insert SQ-XML code of advanced elements to the export specification.

Advanced

List of all the advanced elements in the export

Add
Remove
Up
Down

Select a new advanced element:
- Add a subquery
- Add a projection filtering
- Add a sorting filtering
- Add a grouping
- Add a joining

OK Cancel

Advanced

The remove button deletes selected elements. The Add button creates new elements:

- subquery
- projection filtering
- sorting filtering
- grouping
- joining
Creating a New Export File

Prerequisite

Make sure the Taleo Connect Client application is open.

Steps

1. **File>New>New Export Wizard>Next**

2. Select an exportable entity from one of the available product models.
   a) Select product from pull down menu. The Model is automatically selected based on the product.
b) Select exportable entity from list. A filter is available to narrow down the choices.

3. Select file base.

Create new export is selected by default. This will create an empty export specification.

Templates are available for certain entities. When one of those entities is selected, Create export from a template becomes available. This will create a populated export specification.

a) Select Create export from a template.

b) Select Template from pull down menu.

4. Click Finish.

The New export specification is displayed in the export editor.

**Saving a New Export File**

**Prerequisite**

Make sure a new export specification file is open in export editor.

**Steps**

1. *File* > *Save*
2. Select directory.
3. Enter file name or select existing name in directory.
4. Select file type.
5. Click Save.

**Opening an Existing Export File**

**Prerequisite**

Make sure the Taleo Connect Client application is open.

**Steps**

1. *File* > *Open* > *Export*
2. Select directory.
3. Select file in directory.
4. Click Open.

**Building an Export Specification**

**Steps**

1. Verify that the information in the General tab is correct.
2. Click the Projections tab.
3. Create projections.
   a) Drag and drop the required fields from the entity structure view to the projections list.
   b) Click Add to insert other types of projections.
4. Click the Filters tab.
a) Drag and drop the fields to use as simple filters from the entity structure view to the filters list.
b) Set values, data types, and operator.
c) Click Add to create other types of filters.

5. Click Sorting.
   a) Drag and drop the fields to sort from the entity structure view to the filters list.
   b) Set sorting order.
   c) Click Add to create other types of sorters.

6. Click Advanced.
   a) Click Add to create advanced queries.

7. Save file.
Configuration

Configuration Editor

The Configuration Editor creates the integration process workflow that sends the request file to the Taleo product and retrieves the response file.

The configuration is what defines:

- The overall behavior of the integration process workflow
- The type of monitoring associated to a workflow execution
- The alerts triggered by the workflow execution
- How to handle and transform the source file eventually sent as a the request message to the Taleo environment
- The connectivity information required to contact the Taleo environment
- How to handle and transform the response message received from the Taleo environment
General Tab

The general tab presents information that applies to the whole configuration file.

Workflow

The Workflow identifier should be sufficiently unique to unambiguously define each particular execution instance of the workflow.

The Message type determines the format of the request and response messages exchanged with the Taleo environment. This does not place any restrictions on the original request or ultimate response messages, except for possibly requiring transformations within the workflow.

The Product selected at file creation.

The Communication type defines where the integration instructions are sent within the Taleo environment.
• PRODUCT: The messages are exchanged directly with the Taleo product. This requires a T-SOAP or RS-XML message type.
• MIDDLEWARE: The messages are exchanged with the Taleo middleware. There are no requirements on the message type.
• MIXED: This communication type is deprecated and should not be used. It is present only to support legacy processes.

The T-SOAP format defines the structure of T-SOAP messages. The current version is 2.1.0, version 2.0.0 is available only to support legacy processes.

The temporary files folder defines where the files created during workflow execution are stored. By default, the temporary files are deleted upon workflow completion.

Enable Schedule Monitoring determines the conservation of configuration execution records. The execution records can be viewed in the scheduling monitoring report available from the TCC menu bar Report.

Request/Response
File information for the initial request and ultimate answer.

The Format describes the structure of the message.

• T-SOAP: main message format used by the current Taleo Connect platform, it contains information wrapped in an industry standard SOAP envelope with certain Taleo specific information.
• RS-XML: proprietary Taleo legacy message format similar in nature but not in structure to the T-SOAP format.
• SQ-XML: proprietary Taleo XML format that expresses export specifications.
• XML: any file in the industry standard XML format.
• CSV: flat file format containing records with character delimited values.
• TEXT: any file in a plain text format.
• OTHER: any other file format.

The Encoding states the way the file is encoded. The Taleo environment only supports UTF-8 encoding. Conversion is available in the Pre- and post-processing steps.

The File name can be determined in three ways:

• The file can be provided in the client parameters when invoking a workflow execution at the command line. This option is currently not supported when executing a workflow within the user interface in the runtime perspective.
• The file can be a pre-defined value in which case the file is simply the location on disk of the initial request file.
• The file can be generated using an identifier in which case the folder defines where on disk to resolve the file name.

Monitoring Tab
Taleo Connect Client enables end users to track the progress and inspect the final status of workflow executions by using the monitoring feature. The feature is based on the workflow result information that is updated at each step of the workflow execution. This information can be saved to disk in its original XML format, in a text or HTML formatted version, or in a custom format using an XSL transformation.
**Monitoring**

The Enable Monitoring check box determines if monitoring is active or not. It is highly recommended to always have at least one type of monitoring active as this is the principal source of information about workflow executions.

**Monitoring Location**

Folder identifies the folder where all monitoring files are stored. This value can be modified from its default value; however, when using the HTML monitoring type, the web subfolder is required for the images and style sheets.

A specific monitoring file is generated for each workflow execution and updated after each step. There are two choices for the monitoring file name:

- Use a unique identifier to ensure that all workflow executions create a different monitoring file. This ensures that the monitoring information is always available and easily retrievable for a particular execution.
- Use a constant identifier to have easy access to the result of a given integration process over time. Each execution would overwrite the previous one, but a simple shortcut could be added to an Internet browser to quickly access the current result.

**Monitoring Type**

The Monitoring Type section determines what information is stored and how it is presented.
The XML monitoring file contains all the information produced by the workflow execution. It is not easily readable and is more suited for automated system treatment.

The HTML monitoring file offers a styled view of the main monitoring information that is easy to read and interpret.

The TEXT monitoring file contains the minimal subset of monitoring information and is mostly used in a production environment where an HTML file cannot be accessed.

The Custom monitoring file allows a personalized view of the monitoring information. The custom file is generated by applying an XSL transformation to the XML monitoring file. The file extension determines what is appended to the file location set in the previous location to create the actual file on disk.

**Alerting Tab**

The Alerting feature of Taleo Connect Client (TCC) enables the user to receive e-mail notification of workflow completion or failure.

The TCC Monitoring feature is very helpful to get detailed information about the workflow execution of an integration process during its execution. However, when processes are scheduled to run at times where it is not practical for a user to monitor the execution, an e-mail report can help the user stay informed on the status of the integration processes. The end user can choose to receive e-mails as text or in a custom format using an XSL transformation.

**General**

The mail host SMTP server address is determined by specifying the host alias or the IP address and the server port.
**Completed/Error**

The Completed and Error sections determine if and how an e-mail alert is sent when the workflow is completed or completes with an error, respectively.

To specifies a list of comma-separated e-mail addresses of recipients that will receive the alert e-mail.

CC specifies a list of comma-separated e-mail addresses of recipients that will receive a copy of the alert e-mail.

Subject determines the subject line of the alert e-mail.

The alert e-mail can be provided in two formats:

- Text format is a plain text version of the monitoring information.
- Custom format is built by applying an XSL transformation to the XML monitoring file. The Content-type or the resulting content can also be specified to facilitate e-mail reception processing.

**Pre-processing Tab**

The Pre-processing steps define all handling and transformations to the source file, as well as any other task that needs to be done, prior to sending the request message to the Taleo environment.

**Pre-processing steps**

The Add button creates new steps. The remove button deletes selected step. The Up and Down buttons allows sorting of the steps in the required order.

- Prepare Import
The Prepare Import step converts an initial request message in CSV format to a standard Taleo T-XML import document. The conversion rules are set in the import specification file.

- **Last Run Date Export**
  The Prepare Last Run Date (LRD) step gets the current date and time, and on successful completion of the workflow updates the LRD file. By default, Use date from the zone is selected. Use date from current computer is for extracting from Professional 7.0.4 or earlier.

- **Prepare Export**
  The Prepare Export step adjusts and embeds an initial request message in SQ-XML format to a Taleo standard T-XML export document. The request details and specific configurations are set in the export specification file.

- **XSL based transformation**
  The Transform with XSL step is used to convert an XML file to a valid data file compliant with the defined import specification. It determines XSL file path and name, output format, and encoding. The pre-processing step processes the whole document at once. XSL technology is not designed to support large transformations, the whole file must be loaded in memory prior to conversion. Make sure you have enough available memory to use the XSL transformation preprocessing step with large import files.

- **Wrap Soap**
  The Wrap SOAP step wraps a T-XML document within a T-SOAP envelope to create a message that can be sent to the Taleo environment. This is a mandatory step when using the Prepare Import or Prepare Export steps.

- **Generate ID**
  The Generate ID step determines how to construct the unique message identifier for the T-SOAP message sent to the Taleo environment. If the workflow identifier is unique, then it can be reused; if not, a new identifier can be specified. It is required to use a unicity token within the identifier to ensure uniqueness of the resulting message identifier.

- **Validate Encoding**
  The Validate Encoding step ensures the file is correctly encoded.

- **Convert Encoding**
  The Convert encoding step changes the encoding of the request message.

  Expected encoding: Current type
  Target encoding: Desired type

  A Byte order mark (BOM) byte can be generated if required.

**Custom Steps**

- **Default Custom step**
  Allows implementors to basically execute any logic they want. Java class is the fully qualified name of a class that extends com.taleo.integration.client.step.BaseCustomStep. The custom step can access any number of parameters provided in the configuration file in the form of name/value pairs.

- **CSV Mapping**
  The CSV Mapping step processes the input CSV file columns and maps the values to a new value based on an external mapping file.

  **Standard Parameters**
  
  **Value delimiter**: CSV input file column delimiter character.
  
  **Quotation character**: CSV input file quotation character.
Mapping columns: List of column numbers that represent the first column of each block of columns that are used as a key in the mapping process. Each block number must be separated by a semi-colon (;). The first column in the CSV file is always 1. For example, the key length is 2, the first block of 2 columns starts at column number 4, and the next block starts at column 12, the value in this field must be "4;12" (without the double quotes).

Mapping file: Mapping process definition file. It must contain 1 more column than the number of columns used for the key. The last column in this file contains the value that will be used as a replacement value for the specified column(s). The file must be encoded in UTF-8, must have a CSV header, must use a comma (,) for value delimiter and a double quote (") for the quotation character.

Case sensitive keys: Indicates if the mapping process must be case sensitive for the key(s) comparison.

Advanced parameters

DefaultValue: Default value to use when a key is not found in the mapping file. Provides value to prevent mapping process generating an error by default.

DuplicateKeys: Duplicate keys in the mapping file generate an error by default. Setting the value to "ignore" will use the last defined duplicate key value as replacement in the original CSV file. Possible Values: error, ignore.

Net Change Computation

Compares the customer CSV request file with the Net Change database to determine the changes (create, update, delete, etc...) required to synchronize the target application.

Net Change Initial Load Computation

Creates the Net Change reference database for the selected service (entity). First, it clears the existing entity Net Change tables. Then it extracts data from the target application, based on the Import File specification, to fill the entity Net Change reference table.

FTP Get

Retrieves a file from an FTP server.

Standard Parameters

Host: Name or IP address of the FTP server.

Port: Port number on which the FTP server listens for requests (for example, 4566), The default is 21.

User: The valid user that will be used for the logon on the remote FTP server.

Password: The valid password for the FTP user specified in username that will be used for the logon on the server.

Transfer type: Ftp supports both ASCII and binary image file transfer types. The default is binary. You should use ASCII when transferring text files. In ASCII mode, character conversions to and from the network standard character set are performed. For example, end-of-line characters are converted as necessary, based on the target operating system.

Remote path: Working directory of the FTP server (for example, /tmp/pub). If the directory does not exist, the server throws an exception.

File name: Name of the remote file.

Move file: The request file will be moved into a specific folder determined by Move Folder parameter.
**Move folder:** The move request file destination folder name. If not specified the request file will not be moved.

**Number of retries:** Represents the number of retries that will be done if the operation fails. Zero by default.

**Time between retries:** Represents the number of seconds to wait between the retry attempts. Zero by default.

**Retry factor:** This factor is used as a multiplier on the “Time between retries” between each attempt. It multiplies the last time waited by this amount. i.e. If “Time between retries” is 10 seconds and the retry factor is 2, the first wait time will be 10sec, the second will be 20sec, the third will be 40sec and so on, until the maximum number of retries has been reached.

**Advanced parameters**

**Account:** Optional. The user name for an account on the FTP server (for example, FADA). Specify account if your FTP host requires account information. The account is defined in the FTP protocol to further identify the user that is identified by the username and password input variables.

**Transfer mode:** Ftp supports both of the following file transfer modes: passive or active. The default is passive.

**Data port:** Optional. Listener port number of the data transfer channel. If you do not specify dataport, the FTP Client will throw an exception. This value is used only when the transfer type value is active. In active mode, the FTP client opens a dynamic port, sends the FTP server the dynamic port number on which it is listening over the control stream and waits for a connection from the FTP server. When the FTP server initiates the data connection to the FTP client it binds the source port to port 20 on the FTP server.

**Client host:** Optional. The client host that listen on a port determined by above DataPort. The host name can either be a machine name, such as “www.taleo.com”, or a textual representation of its IP address. If a literal IP address is supplied, only the validity of the address format is checked.

**Timeout:** Optional. Sets the timeout in milliseconds to use when reading from the data connection. This timeout will be set immediately after opening the data connection.

**Create remote path if missing:** Create the working directory on the FTP server (for example, /tmp/pub/a/b).

- **FTP Put**

  Sends a file to an FTP server.

**Standard Parameters**

**Host:** Name or IP address of the FTP server.

**Port:** Port number on which the FTP server listens for requests (for example, 4566), The default is 21.

**User:** The valid user that will be used for the logon on the remote FTP server.

**Password:** The valid password for the FTP user specified in username that will be used for the logon on the server.

**Transfer type:** Ftp supports both ASCII and binary image file transfer types. The default is binary. You should use ASCII when transferring text files. In ASCII mode, character conversions to and from the network standard character set are performed. For example, end-of-line characters are converted as necessary, based on the target operating system.

**Remote path:** Working directory of the FTP server (for example, /tmp/pub). If the directory does not exist, the server throws an exception.
**File name**: Name of the remote file.

**Number of retries**: Represents the number of retries that will be done if the operation fails. Zero by default.

**Time between retries**: Represents the number of seconds to wait between the retry attempts. Zero by default.

**Retry factor**: This factor is used as a multiplier on the “Time between retries” between each attempt. It multiplies the last time waited by this amount. i.e. If “Time between retries” is 10 seconds and the retry factor is 2, the first wait time will be 10sec, the second will be 20sec, the third will be 40sec and so on, until the maximum number of retries has been reached.

**Advanced parameters**

**Account**: Optional. The user name for an account on the FTP server (for example, FADA). Specify account if your FTP host requires account information. The account is defined in the FTP protocol to further identify the user that is identified by the username and password input variables.

**Transfer mode**: Ftp supports both of the following file transfer modes: passive or active. The default is passive.

**Data port**: Optional. Listener port number of the data transfer channel. If you do not specify dataport, the FTP Client will throw an exception. This value is used only when the transfer type value is active. In active mode, the FTP client opens a dynamic port, sends the FTP server the dynamic port number on which it is listening over the control stream and waits for a connection from the FTP server. When the FTP server initiates the data connection to the FTP client it binds the source port to port 20 on the FTP server.

**Client host**: Optional. The client host that listen on a port determined by above DataPort. The host name can either be a machine name, such as "www.taleo.com", or a textual representation of its IP address. If a literal IP address is supplied, only the validity of the address format is checked.

**Timeout**: Optional. Sets the timeout in milliseconds to use when reading from the data connection. This timeout will be set immediately after opening the data connection.

**Create remote path if missing**: Create the working directory on the FTP server (for example, /tmp/pub/a/b).

- **FTP Delete**
  Removes a file from an FTP server.

**Standard Parameters**

**Host**: Name or IP address of the FTP server.

**Port**: Port number on which the FTP server listens for requests (for example, 4566), The default is 21.

**User**: The valid user that will be used for the logon on the remote FTP server.

**Password**: The valid password for the FTP user specified in username that will be used for the logon on the server.

**Remote path**: Working directory of the FTP server (for example, /tmp/pub). If the directory does not exist, the server throws an exception.

**File name**: Name of the remote file.

**Number of retries**: Represents the number of retries that will be done if the operation fails. Zero by default.
Time between retries: Represents the number of seconds to wait between the retry attempts. Zero by default.

Retry factor: This factor is used as a multiplier on the "Time between retries" between each attempt. It multiplies the last time waited by this amount. i.e. If "Time between retries" is 10 seconds and the retry factor is 2, the first wait time will be 10sec, the second will be 20sec, the third will be 40sec and so on, until the maximum number of retries has been reached.

Advanced parameters

Account: Optional. The user name for an account on the FTP server (for example, FADA). Specify account if your FTP host requires account information. The account is defined in the FTP protocol to further identify the user that is identified by the username and password input variables.

Transfer mode: Ftp supports both of the following file transfer modes: passive or active. The default is passive.

Data port: Optional. Listener port number of the data transfer channel. If you do not specify dataport, the FTP Client will throw an exception. This value is used only when the transfer type value is active. In active mode, the FTP client opens a dynamic port, sends the FTP server the dynamic port number on which it is listening over the control stream and waits for a connection from the FTP server. When the FTP server initiates the data connection to the FTP client it binds the source port to port 20 on the FTP server.

Client host: Optional. The client host that listen on a port determined by above DataPort. The host name can either be a machine name, such as "www.taleo.com", or a textual representation of its IP address. If a literal IP address is supplied, only the validity of the address format is checked.

Timeout: Optional. Sets the timeout in milliseconds to use when reading from the data connection. This timeout will be set immediately after opening the data connection.

- HTTP

Posts data to an HTTP/HTTPS URL.

Standard Parameters

Endpoint: Represents the endpoint to use for the post operation. This endpoint must use the HTTP or HTTPS protocol.

Number of retries: Represents the number of retries that will be done if the operation fails. Zero by default.

Time between retries: Represents the number of seconds to wait between the retry attempts. Zero by default.

Retry factor: This factor is used as a multiplier on the "Time between retries" between each attempt. It multiplies the last time waited by this amount. i.e. If "Time between retries" is 10 seconds and the retry factor is 2, the first wait time will be 10sec, the second will be 20sec, the third will be 40sec and so on, until the maximum number of retries has been reached.

Advanced parameters

Custom HTTP headers are supported through the use of advanced parameters. A parameter having the prefix Header: followed by the name of the header as the name of the parameter can be defined. For example, to define the Content-type the parameter name will be Header:Content-type.

- SMTP

Sends import/export result file by email.

Standard Parameters

Host: The address of the host SMTP server.
Port: The port of the host SMTP server. 25 by default.

From: The address of the sender. Taleo Connect Client by default.

To: The address(es) of the receiver(s).

Cc: The address(es) of the receiver(s).

Subject: The subject of the email.

MessageTemplate: The template string that contains the email message body. The template supports the following velocity standard tools: DateTool and MathTool. See the Apache velocity website for details. It also supports the following Taleo custom velocity tools: StringTool and FileTool. The StringTool variables are: $string.rightPad("123", 10), $string.rightPad("123", 10, "-_"), $string.leftPad("123", 10) and $string.leftPad("123", 10, "0"). The FileTool variables are: $file.content(), $file.lineCount() and $file.size().

Number of retries: Represents the number of retries that will be done if the operation fails. Zero by default.

Time between retries: Represents the number of seconds to wait between the retry attempts. Zero by default.

Retry factor: This factor is used as a multiplier on the “Time between retries” between each attempt. It multiplies the last time waited by this amount. i.e. If “Time between retries” is 10 seconds and the retry factor is 2, the first wait time will be 10sec, the second will be 20sec, the third will be 40sec and so on, until the maximum number of retries has been reached.

SendAttachment: Send the step input file as attachment. True by default.

ZipAttachment: Compress the attached file.

Attachment file name: Enter name for attached file. The file name MUST NOT have leading or trailing spaces. The file name MUST NOT contain the \\ / * ? < > | characters. If left empty, the default file name will be used.

Send Notification: Sends notification email on selected condition. Condition is determined by processing status provided by previous step. Select “Always” to disregard previous step processing status and send notification on all step executions. Select “On warning or error” to send a notification only if the previous step processing status is “warning” or “error”. Select “On error” to send a notification only if the previous step processing status is “error”. The possible values are: Always, On warning or error, On error.
Send Tab

The Send processing step defines the connectivity information required to contact the Taleo environment.

Endpoint Source

Endpoint information can originate from the default configuration board file, the configuration file board tab, or from the current step.

Endpoint

The Endpoint section provides the actual connectivity information for the Taleo environment:

- Protocol: HTTP or HTTPS
- Host: Alias or IP address
- Port: Number

The communication path can be provided by the full Path directly or by a combination of the service dispatcher Path and Service Name. The complete web service path can also be provided.

Authentication

The Authentication section provides the authentication context for the requests. User/password HTTP Basic Authentication is required to connect to the Taleo environment. No authentication is listed as an option for testing purposes.

Quality

The Quality section determines how communication failures are handled within the workflow execution.
Retry determines what action is taken when a particular processing operation fails. This option is optional for the send step.

- Interval determines the number of seconds between each try.
- Acknowledgement Attempts is a deprecated option only kept for backwards compatibility reasons and should not be used.
- Completion attempts is the number of times the request file will be sent to the zone before halting the workflow and generating an error message.
- Ignore errors determines if retries are attempted even if an error has occurred. This option is mandatory when using retries in the send step as the only two possible outcomes are success or error.

Socket time-out determines how much time a particular processing operation will wait for a response. This is a highly technical setting and only needs to be modified in particular cases where connectivity issues occur with the Taleo environment.

**Middleware**

The Middleware section provides extra information required when communicating with the Taleo middleware. These settings are only enabled if the communication type is set to MIDDLEWARE.

Sender ID, Receiver ID and Event ID specify information required for the routing of the request message within the middleware.

**Miscellaneous**

The Miscellaneous section sets some very technical send step information.

Chunked limit affects how large request messages are sent using the HTTP protocol.

Validate request is a legacy option only offered in RS-XML workflows where the request message was pre-validated.

**Preview**

Preview the send configuration connectivity information defined in the endpoint section.
Poll Tab

The Poll processing step defines the connectivity information required to contact the Taleo environment.

Endpoint Source

Endpoint information can originate from the default configuration board file, the configuration file board tab, or from the current step.

Endpoint

The Endpoint section provides the actual connectivity information for the Taleo environment:

- Protocol: HTTP or HTTPS
- Host: Alias or IP address
- Port: Number

The communication path can be provided by the full Path directly or by a combination of the service dispatcher Path and Service Name. The complete web service path can also be provided.
**Authentication**

The Authentication section provides the authentication context for the requests. User/password HTTP Basic Authentication is required to connect to the Taleo environment. No authentication is listed as an option for testing purposes.

**Quality**

The Quality section determines how communication failures are handled within the workflow execution. Retry determines what action is taken when a particular processing operation fails. This option is mandatory for the poll step. An integration process will very rarely complete on the first polling attempt.

- Interval determines the number of seconds between each try.
- Acknowledgement Attempts is a deprecated option only kept for backwards compatibility reasons and should not be used.
- Completion attempts is the number of times the zone will be polled for the result file before halting the workflow and generating an error message. The completion attempt counter automatically resets when the zone processed record count changes.
- Ignore errors determines if retries are attempted even if an error has occurred.

Socket time-out determines how much time a particular processing operation will wait for a response. This is a highly technical setting and only needs to be modified in particular cases where connectivity issues occur with the Taleo environment.

**Preview**

Preview the poll configuration connectivity information defined in the endpoint section.
Retrieve Tab

The Retrieve processing step defines the connectivity information required to contact the Taleo environment.

**Retrieve**

**Endpoint Source**
Select the connectivity source information
- Use an endpoint from the list
  - Endpoint: ACME
- Provided in the current step

**Endpoint**
Configure the connectivity information
- Protocol: HTTPS
- Host: acme.taleo.net
- Port: 443
- Use default service dispatcher path
- Provide the service dispatcher path
  - Path: 
  - Service name: 
- Provide the complete web service path
  - Path: 

**Quality**

**Miscellaneous**

**Preview**
Preview the end result of the retrieve configuration
- URL: https://acme.taleo.net:443/smartorg/ServiceDispatcher?ServiceName=IntegrationManagementService

**Endpoint Source**
Endpoint information can originate from the default configuration board file, the configuration file board tab, or from the current step.

**Endpoint**
The Endpoint section provides the actual connectivity information for the Taleo environment:
- Protocol: HTTP or HTTPS
- Host: Alias or IP address
- Port: Number

The communication path can be provided by the full Path directly or by a combination of the service dispatcher Path and Service Name. The complete web service path can also be provided.

**Authentication**
The Authentication section provides the authentication context for the requests. User/password HTTP Basic Authentication is required to connect to the Taleo environment. No authentication is listed as an option for testing purposes.

**Quality**
The Quality section determines how communication failures are handled within the workflow execution.
Retry determines what action is taken when a particular processing operation fails. This option is optional for retrieve step.

- Interval determines the number of seconds between each try.
- Acknowledgement Attempts is a deprecated option only kept for backwards compatibility reasons and should not be used.
- Completion attempts is the number of times to try and retrieve the result file from the zone before halting the workflow and generating an error message.
- Ignore errors determines if retries are attempted even if an error has occurred. This option is mandatory when using retries in the retrieve step as the only two possible outcomes are success or error.

Socket time-out determines how much time a particular processing operation will wait for a response. This is a highly technical setting and only needs to be modified in particular cases where connectivity issues occur with the Taleo environment.

**Middleware**

The Middleware section of the retrieve step provides extra information required when communicating with the Taleo middleware. These settings are only enabled if the communication type is set to MIDDLEWARE.

Sender ID, Receiver ID and Event ID specify information required for the routing of the request message within the middleware.

**Miscellaneous**

The Miscellaneous section sets some very technical send step information.

Chunked limit affects how large request messages are sent using the HTTP protocol.

Validate request is a legacy option only offered in RS-XML workflows where the request message was pre-validated.

**Preview**

Preview the retrieve configuration connectivity information defined in the endpoint section.
Post-processing Tab

The Post-processing steps define how to handle and transform the source file eventually sent as the response message from the Taleo environment.

Post-processing steps

The Up and Down buttons allows sorting of the steps in the required order. The remove button deletes selected step. The Add button creates new steps:

- **Strip SOAP**
  
  Determines how the SOAP envelope will be removed from the response message:
  
  - Document: leaves an XML message with the content and the attributes of the document.
  - Content: leaves an XML message with the business content of the document.
  - CSV: leaves a CSV message with the business content of the document, applies to CSV mode export responses only.
  
  The Fail on export error stops processing if the response message contains an error.

- **XSL based transformation**
  
  Applies an XSL transformation to a response message. This step works at the record level. The template file must contain a placeholder ([RECORD_OUTPUT]) that will contain the transformed data.
  
  XSL file: Specifies the path and file name to use.
Template file: Specifies the path and file name to use.

Export mode: File format before transformation T-XML or XML

Output format: File format after transformation, OTHER, TEXT, CSV, XML, SQ-XML, T-XML, T-SOAP, RS-XML

**Import Result Transformation**

Converts the import results to a CSV file using the following specifications.

File: Specifies the path and file name to use.

Output properties: When selected, inserts operation properties at beginning of result file.

Advanced Parameters are available to add data columns to the CSV result file.

EXTRA_COLUMN_NAME_x: The parameter value sets the column header name.

EXTRA_COLUMN_XPATH_x: The parameter value sets the Xpath for the data inserted into the column. The Xpath for a particular set of data can be found in the T-XML response file returned by the operation. Since the response document contains a namespace, the "local name" must be specified. For example, if the Xpath in the T-XML file is //Candidate/Number, the parameter value is expressed as //*[local-name()="Candidate"]/*[local-name()="Number"].

**Note:** Make sure to replace the x at the end of the pairs of parameter names with the same number to ensure they remain matched. For example, EXTRA_COLUMN_NAME_1 and EXTRA_COLUMN_XPATH_1, EXTRA_COLUMN_NAME_2 and EXTRA_COLUMN_XPATH_2, etc...

**Export Result Transformation**

Converts the exports results to a CSV file using the export specifications

**Convert Encoding**

Changes the encoding of the response message if necessary.

Expected encoding: Current type

Target encoding: Desired type

A Byte order mark (BOM) byte can be generated if required.

**Custom Steps**

**Default Custom step**

Allows implementors to basically execute any logic they want. Java class is the fully qualified name of a class that extends com.taleo.integration.client.step.BaseCustomStep. The custom step can access any number of parameters provided in the configuration file in the form of name/value pairs.

**CSV Mapping**

The CSV Mapping step processes the input CSV file columns and maps the values to a new value based on an external mapping file.

**Standard Parameters**

Value delimiter: CSV input file column delimiter character.

Quotation character: CSV input file quotation character.

Mapping columns: List of column numbers that represent the first column of each block of columns that are used as a key in the mapping process. Each block number must be separated by a semi-colon (;). The first column in the CSV file is always 1. For example, the key length is 2, the first block of 2 columns starts at column number 4, and the next block starts at column 12, the value in this field must be "4;12" (without the double quotes).
**Mapping file:** Mapping process definition file. It must contain 1 more column than the number of columns used for the key. The last column in this file contains the value that will be used as a replacement value for the specified column(s). The file must be encoded in UTF-8, must have a CSV header, must use a comma (,) for value delimiter and a double quote (") for the quotation character.

**Case sensitive keys:** Indicates if the mapping process must be case sensitive for the key(s) comparison.

**Advanced parameters**

**DefaultValue:** Default value to use when a key is not found in the mapping file. Provides value to prevent mapping process generating an error by default.

**DuplicateKeys:** Duplicate keys in the mapping file generate an error by default. Setting the value to "ignore" will use the last defined duplicate key value as replacement in the original CSV file. Possible Values: error, ignore.

- **Net Change Result Processor**
  Generates a simple XML report that indicates how many entities have been processed during the operation.

- **Net Change Result Transformation**
  Processes the file generated by the "Net Change Result Processor" to transform it into a standard TCC CSV file. This step is optional for Initial Load processing.

- **FTP Put**
  Sends a file to an FTP server.

**Standard Parameters**

**Host:** Name or IP address of the FTP server.

**Port:** Port number on which the FTP server listens for requests (for example, 4566). The default is 21.

**User:** The valid user that will be used for the logon on the remote FTP server.

**Password:** The valid password for the FTP user specified in username that will be used for the logon on the server.

**Transfer type:** Ftp supports both ASCII and binary image file transfer types. The default is binary. You should use ASCII when transferring text files. In ASCII mode, character conversions to and from the network standard character set are performed. For example, end-of-line characters are converted as necessary, based on the target operating system.

**Remote path:** Working directory of the FTP server (for example, /tmp/pub). If the directory does not exist, the server throws an exception.

**File name:** Name of the remote file.

**Number of retries:** Represents the number of retries that will be done if the operation fails. Zero by default.

**Time between retries:** Represents the number of seconds to wait between the retry attempts. Zero by default.

**Retry factor:** This factor is used as a multiplier on the "Time between retries" between each attempt. It multiplies the last time waited by this amount. i.e. If "Time between retries" is 10 seconds and the retry factor is 2, the first wait time will be 10sec, the second will be 20sec, the third will be 40sec and so on, until the maximum number of retries has been reached.
**Account:** Optional. The user name for an account on the FTP server (for example, FADA). Specify account if your FTP host requires account information. The account is defined in the FTP protocol to further identify the user that is identified by the username and password input variables.

**Transfer mode:** Ftp supports both of the following file transfer modes: passive or active. The default is passive.

**Data port:** Optional. Listener port number of the data transfer channel. If you do not specify dataport, the FTP Client will throw an exception. This value is used only when the transfer type value is active. In active mode, the FTP client opens a dynamic port, sends the FTP server the dynamic port number on which it is listening over the control stream and waits for a connection from the FTP server. When the FTP server initiates the data connection to the FTP client it binds the source port to port 20 on the FTP server.

**Client host:** Optional. The client host that listen on a port determined by above DataPort. The host name can either be a machine name, such as "www.taleo.com", or a textual representation of its IP address. If a literal IP address is supplied, only the validity of the address format is checked.

**Timeout:** Optional. Sets the timeout in milliseconds to use when reading from the data connection. This timeout will be set immediately after opening the data connection.

**Create remote path if missing:** Create the working directory on the FTP server (for example, /tmp/pub/a/b).

- **FTP Delete**

  Removes a file from An FTP server.

**Standard Parameters**

**Host:** Name or IP address of the FTP server.

**Port:** Port number on which the FTP server listens for requests (for example, 4566), The default is 21.

**User:** The valid user that will be used for the logon on the remote FTP server.

**Password:** The valid password for the FTP user specified in username that will be used for the logon on the server.

**Remote path:** Working directory of the FTP server (for example, /tmp/pub). If the directory does not exist, the server throws an exception.

**File name:** Name of the remote file.

**Number of retries:** Represents the number of retries that will be done if the operation fails. Zero by default.

**Time between retries:** Represents the number of seconds to wait between the retry attempts. Zero by default.

**Retry factor:** This factor is used as a multiplier on the "Time between retries" between each attempt. It multiplies the last time waited by this amount. i.e. If "Time between retries" is 10 seconds and the retry factor is 2, the first wait time will be 10sec, the second will be 20sec, the third will be 40sec and so on, until the maximum number of retries has been reached.

**Advanced parameters**

**Account:** Optional. The user name for an account on the FTP server (for example, FADA). Specify account if your FTP host requires account information. The account is defined in the FTP protocol to further identify the user that is identified by the username and password input variables.

**Transfer mode:** Ftp supports both of the following file transfer modes: passive or active. The default is passive.
**Data port:** Optional. Listener port number of the data transfer channel. If you do not specify dataport, the FTP Client will throw an exception. This value is used only when the transfer type value is active. In active mode, the FTP client opens a dynamic port, sends the FTP server the dynamic port number on which it is listening over the control stream and waits for a connection from the FTP server. When the FTP server initiates the data connection to the FTP client it binds the source port to port 20 on the FTP server.

**Client host:** Optional. The client host that listen on a port determined by above DataPort. The host name can either be a machine name, such as "www.taleo.com", or a textual representation of its IP address. If a literal IP address is supplied, only the validity of the address format is checked.

**Timeout:** Optional. Sets the timeout in milliseconds to use when reading from the data connection. This timeout will be set immediately after opening the data connection.

- **HTTP**
  Posts data to an HTTP/HTTPS URL.

**Standard Parameters**

**Endpoint:** Represents the endpoint to use for the post operation. This endpoint must use the HTTP or HTTPS protocol.

**Number of retries:** Represents the number of retries that will be done if the operation fails. Zero by default.

**Time between retries:** Represents the number of seconds to wait between the retry attempts. Zero by default.

**Retry factor:** This factor is used as a multiplier on the "Time between retries" between each attempt. It multiplies the last time waited by this amount. i.e. If "Time between retries" is 10 seconds and the retry factor is 2, the first wait time will be 10sec, the second will be 20sec, the third will be 40sec and so on, until the maximum number of retries has been reached.

**Advanced parameters**

Custom HTTP headers are supported through the use of advanced parameters. A parameter having the prefix Header: followed by the name of the header as the name of the parameter can be defined. For example, to define the Content-type the parameter name will be Header:Content-type.

- **SMTP**
  Sends import/export result file by email.

**Standard Parameters**

**Host:** The address of the host SMTP server.

**Port:** The port of the host SMTP server. 25 by default.

**From:** The address of the sender. Taleo Connect Client by default.

**To:** The address(es) of the receiver(s).

**Cc:** The address(es) of the receiver(s).

**Subject:** The subject of the email.

**MessageTemplate:** The template string that contains the email message body. The template supports the following velocity standard tools: DateTool and MathTool. See the Apache velocity website for details. It also supports the following Taleo custom velocity tools: StringTool and FileTool. The StringTool variables are: $string.rightPad("123", 10), $string.rightPad("123", 10, "_"), $string.leftPad("123", 10) and $string.leftPad("123", 10, "0"). The FileTool variables are: $file.content(), $file.lineCount() and $file.size().
**Number of retries**: Represents the number of retries that will be done if the operation fails. Zero by default.

**Time between retries**: Represents the number of seconds to wait between the retry attempts. Zero by default.

**Retry factor**: This factor is used as a multiplier on the “Time between retries” between each attempt. It multiplies the last time waited by this amount. i.e. If “Time between retries” is 10 seconds and the retry factor is 2, the first wait time will be 10sec, the second will be 20sec, the third will be 40sec and so on, until the maximum number of retries has been reached.

**SendAttachment**: Send the step input file as attachment. True by default.

**ZipAttachment**: Compress the attached file.

**Attachment file name**: Enter name for attached file. The file name MUST NOT have leading or trailing spaces. The file name MUST NOT contain the \ / : * ? < > | characters. If left empty, the default file name will be used.

**Send Notification**: Sends notification email on selected condition. Condition is determined by processing status provided by previous step. Select “Always” to disregard previous step processing status and send notification on all step executions. Select “On warning or error” to send a notification only if the previous step processing status is “warning” or “error”. Select “On error” to send a notification only if the previous step processing status is “error”. The possible values are: Always, On warning or error, On error.

- **Report Filtering**
  Filters record results by criteria.

  **reportFiltering**: Select from drop down list. All (no filter), On error or warning, On error only.

**Board Tab**

The Board Tab allows the user to create endpoint and symbol elements specific to the current configuration file.

![Board Tab Image]
With specific endpoints and symbols the user can override the settings of the default configuration board.

For additional information on endpoints and symbol elements refer to the Default Configuration Board section.

Creating a New Configuration File

**Prerequisite**
Make sure the Taleo Connect Client application is open.

**Steps**
1. **File>New>New Configuration Wizard>Next**
2. Select the configuration file base.
   a) Click on navigate.
   b) Select directory.
   c) Select file.
   d) Click Open.
3. Click Next.
4. Verify connectivity information.
5. Click Finish.

The New configuration specification is displayed in the configuration editor.

Saving a New Configuration File

**Prerequisite**
Make sure a new configuration specification file is open in configuration editor.

**Steps**
1. **File>Save**
2. Select directory.
3. Enter file name or select existing name in directory.
4. Select file type.
5. Click Save.

Opening an Existing Configuration File

**Prerequisite**
Make sure the Taleo Connect Client application is open.

**Steps**
1. **File>Open>Configuration**
2. Select directory.
3. Select file in directory.
4. Click Open.

**Executing a Configuration**

**Prerequisite**
Make sure a configuration file is open and active.

**Steps**
Click Execute the configuration.
Runtime Perspective

The Runtime perspective displays the workflow execution details in graphical and tabular formats.

Workflow Identifier
The workflow identifier for the current execution displays above the Workflow Execution Details icons.

Workflow Execution Details
A graphical overview of the step progress groups all processing steps in a single aggregated icon.

Basic Information
The basic monitoring information can be seen in the left hand column and consists of the overall state of the workflow execution and the individual step statuses.

- If the workflow execution is completed, then all steps will also be completed.
- If the workflow execution is in progress, then a subset of all steps will be completed.
• If the workflow execution is in error, then a particular step may also be in error. However, certain error scenarios occur at the initialization or finalization of the workflow execution, hence either no steps or all steps will have been completed.

General Information
General information is provided and consists of:

1. Workflow type, start, finish, and duration of the execution (when applicable).
2. The original location of the request message file and the target location of the response message file.

The presence of the response message target location DOES NOT imply that the file was actually created. This is only the case if the workflow execution completed will success.

Step Details
The step details tabular view shows the start, finish, duration (when applicable), and the request and response message information when applicable.

Monitoring a Workflow

Prerequisite
A workflow configuration file is available for execution.

Steps
1. Window>Open Perspective>Runtime
   The Runtime Perspective opens.

2. File>Open Configuration
   a) Select directory.
   b) Select file.
   c) Click Open.
   The configuration file opens.

3. Click Execute the configuration

4. Maximise the Runtime Perspective.

Result
The workflow execution details are displayed.
**Console**

**Console Perspective**

The Console perspective displays a real-time view of all integration processes being handled by a Taleo product.

The console perspective uses a single editor, the message list viewer, that lists all the messages in a Taleo product matching a particular criteria. This perspective distinguishes between the message and the document contained in the message. This perspective offers multiple views that analyze each message from different angles.

**Message List Viewer**

The Message List Viewer is an Editor that primarily shows a list of messages from the target Taleo Connect Server. Technically, it is an Editor, since the profile of the target Taleo Connect Server can be edited and saved here.

The information available in the message list is:

- **Identifier**: A universally unique external identifier for the message.
- **Type**: Determines if the message is a request or a response.
• **State**: The current position of the message in the Taleo Connect Server processing workflow.

• **Creation Date**: Indicates the date and time the message was received or created by the Taleo Connect Server.

The three actions available in the Message List viewer are:

• Refresh the current list of messages.

• Filter the list of messages according to a set of criteria.

• Edit the profile of the target Taleo Connect Server.

**Filters**

The Filters define which messages will be part of the displayed list. The available filters are:

• **Key**: Filters the messages based on whether the key matches the exact value provided. Since the message key is a unique value within a Taleo Connect Server instance, specifying a value here guarantees that a single message will be displayed.

• **Identifier**: Filters the messages based on whether the identifier contains the value provided. For example, using the value ‘Candidate’ would match ‘CandidateImport-2006-03-12’ and ‘Extract-Candidates-2005-12-12’.

• **Conversation**: Filters the messages based on whether the conversation contains the value provided. This is the same behavior as the Identifier filter.

• **Target**: Filters the messages based on whether they are requests or responses. The Request option is an alias for the Application target. The Response option includes the Middleware, External, and Queue targets.

• **State**: Filters the messages based on their state in the Taleo Connect Server workflow. The Incomplete option matches all states except In Progress.

• **Date**: Filters the messages based on their creation date in the Taleo Connect Server. Selecting a specific date shows all messages from 00:00:00 to 23:59:59, local time of the computer running the Taleo Connect Console. Selecting a range of dates shows all messages from 00:00:00 of the ‘from’ date to 23:59:59 of the ‘to’ date, local time of the computer running the Taleo Connect Console.

**Profile**

A Profile defines the connectivity information required to access the Taleo Connect Server embedded in a target Taleo product. The general information of a profile is:

• **Description**: Information with no functional significance. May be used by customers to distinguish or identify profiles.

• **Version**: The version of the target Taleo Connect Server. This value determines the communication protocol and the available features.

The connectivity information of a profile is:

• **Protocol**: The type of HTTP communication used: plain HTTP or secure HTTPS.

• **Host**: The host alias or the IP address of the target Taleo product.

• **Port**: The port number of the target Taleo product.

• **Dispatcher**: The location of the target Taleo product web service dispatcher entry point. For the Taleo Professional product this is /servlets/ServiceDispatcher?ServiceName=. For other Taleo products, the value is /product name/ServiceDispatcher?ServiceName=.

• **Username**: The username required to authenticate web service requests to the Taleo Connect Server.

• **Password**: The password required to authenticate web service requests to the Taleo Connect Server.
Message Details

The Message Details View shows the top level information of the integration message.

- **Key**: An internal identifier used by the Taleo Connect Server embedded in a target Taleo product. The message key's only external use is to uniquely associate request and response message instances. The key values are only unique within a particular Taleo product.

- **Identifier**: A universally unique external identifier for the message. It is used by external systems to link messages within larger business processes.

- **Conversation**: An identifier that loosely links several messages together in a business process. This is not a transaction identifier in the usual sense of the word (including two-phase commit and integrity concerns). A conversation is only used for monitoring purposes. In particular, a response generated by a Taleo Connect Server will always have the same conversation identifier as the request message (if none was provided, then the conversation identifier of the response will be the message identifier of the request).

- **Target**: Determines the processing target in the Taleo Connect Server.
  - Application indicates that this message contains a request and is to be processed by the Taleo product.
  - Middleware or External indicates that this message contains a response and is to be pushed to another application.
  - Queue also indicates that the message contains a response, but that the message should remain within the Taleo product and will be polled by an external party.

- **State**: Either a Request or a Response. An integration message is a generic transmission from a source to a target application. The concept of request and response is only linked to the business document contained within the message.

- **Service**: The unique identifier of the processing strategy of the Taleo Connect Server. This value only apply to messages with an Application target, i.e. request messages.

The full raw message content is available using the show full message action.

Message Actions

Under certain specific circumstances, it is possible to interact with messages from the console.

- **Suspend** halts the processing of a message. This action is only applicable to request messages that are in the In Progress state. A signal will be sent to the Taleo product and the message will be marked as Suspending; the actual suspension is dependent on the type of processing and may not be immediate.

- **Resume** resumes the processing of a message. This action is only applicable to messages that are in the Suspended or Interrupted states. The result of this action is to mark the message as Ready so the Taleo Connect Server will process it through the normal message queue. In the case of a request message, the processing will start after the last complete record. In the particular case of an export request, as there is only one record, the entire export is restarted. For response messages, the resume action resends the response to the middleware system when required.

Message Errors

When a technical process error occurs, the details of the problem are available in the second section of the view. It is important to note that only internal processing errors are visible in this view and specifically only when messages are either in the Interrupted or In Error states. Any business errors that occur during the processing of a particular record are stored in the response document and do not affect the overall processing of the message.
• **Source**: Either the Server or the Client. An integration message is a generic transmission from a source to a target application. The concept of request and response is only linked to the business document contained within the message.

• **Code**: An external unique identifier for the type of problem that occurred.

• **Reason**: A contextualized message giving more information about the problem that occurred.

• **Detail**: Further technical information about the problem. Specifically, if the problem was caused by an unmanaged Java language exception, this detail would be the stack trace of the exception.

**Document Detail**

The **Document Detail View** shows the top level information of the integration document.

• **Action**: Defines how the target application is to process the records. The three possible values are Export, Import, or Business Service. In the two first actions, the records are handled by the generic integration services provided by the Taleo Connect Client. The last action signifies that the records are delegated to product specific business services for individual processing.

• **Progress**: Shows how many records have currently been processed. In most cases, this value is identical for both the request and response message. For exports, the request document has a single record and the progress is only significant for the response document where the total indicates the number of entities that will be exported.

Each record processed during the integration is assigned a status: success or error. The total section first shows the overall number of records in the document then the amount of successful records and the amount of records in error.

**Document Content**

The **Document Content View** can display the content of the document or save it to file.

### Creating a Console Profile

**Steps**

1. **File>New>New Profile Wizard>Next**

2. Select the profile type.

   • **Endpoint**: Create the profile based on an endpoint. Select product from pull down menu. Select Endpoint from pull down menu

   • **Blank profile**: Create a blank profile and fill the connectivity information manually.

   • **Demo profile**: Create an off-line demo profile to explore the console features.

3. Click Next

4. Verify or complete connectivity information.

5. Click Finish.

The New console profile is displayed in the console perspective Message list viewer.

### Saving a Console Profile

**Prerequisite**

Make sure a new console profile is open in the message viewer.
Steps
1. **File>Save**
2. Select directory.
3. Enter file name or select existing name in directory.
4. Select file type.
5. Click Save.
Command Line

Script Execution

Taleo Connect Client workflow execution in a production environment is run by a command line script. The command line script can be scheduled to run automatically.

There are three possible command line scripts to launch TCC execution. They are the same for all operating systems. The only difference being that Windows uses .bat files and Unix/Linux uses .sh files. The following files can be specified in a command line script:

- The TCC batch file that is included in the installation package.
  TaleoConnectClient.bat
- An import or export configuration file.
  <touchpoint>_<action>cfg.xml
- An import or export request file.
  <touchpoint>_<action>_ld.xml
  <touchpoint>_<action>_sq.xml
- An import or export result file.
  <touchpoint>_<action>_result_<date_time>.csv
  <touchpoint>_<action>_result.xml
  <touchpoint>_<action>_result_<date_time>.txt
  filename.other

The three possible command line scripts will have the following layout. The path is the absolute or relative location of the specified file.

- "path\TaleoConnectClient.bat" "path\configuration file" "path\request file" "path\result file"
  
  The first script is the typical use for Taleo Connect Client where both the request and response message files are specified explicitly. The workflow configuration usually contains only communication information that applies to a variety of different messages.
- "path\TaleoConnectClient.bat" "path\configuration file" "path\request file"
  
  The second script is typical of a scheduled integration process where the request message varies and the response message file is generated based on the timestamp.
- "path\TaleoConnectClient.bat" "path\configuration file"
  
  The third script is typical of a specialized workflow with extensive pre and/or post-processing that only applies to a single integration process. In this situation, the request and response files are often known and can thus be specified in the configuration itself.
Scheduling Tasks

Scheduling tasks automates the Taleo Connect Client ongoing import and export processes. Scheduled imports and exports are programmed and maintained by users, on scheduler software running in the environment (Windows, Unix, Linux).

Unix/Linux use Cron to schedule tasks. Refer to your system documentation to implement scheduling under Unix/Linux.

Scheduling in Windows

Prerequisite
Integration scripts that have been tested and are ready to scheduled.

Steps
1. Click Start > Programs > Accessories > System Tools > Scheduled Tasks.
   The Scheduled Tasks window opens.
2. Double-click Add Scheduled Task.
   The Scheduled Task Wizard opens.
3. Click Next.
   A file selection window opens in the Wizard.
4. Click Browse...
   The Select Program to Schedule browser window opens.
5. Navigate to the folder where the batch file to schedule is located.
   a) Click on the batch file to select it.
   b) Click Open
6. Enter task name (optional).
   If the batch file name is well constructed and reflects the specific task being automated, you might as well leave the task name as is.
   a) Select task frequency.
   b) Click Next>
   c) Select task frequency details.
   d) Click Next>
   e) Enter Windows domain user name
   f) Enter password
   g) Confirm password
   h) Click Next>
7. Click Finish
   The task is displayed in the Scheduled Tasks window.
Execution from Java

Execution of a Taleo Connect Client workflow from within another Java application is possible. The host application must match the environment constraints defined in the `TaleoConnectClient.bat` script and invoke the `com.taleo.integration.client.Client.execute()` method.

For Taleo Connect Client to execute correctly, the proper execution environment must be available. The following execution script is the minimal required configuration for a typical Windows installation:

```bash
SET IC_HOME="C:\Program Files\Taleo Connect Client";
SET CLASSPATH="%IC_HOME%\lib\taleo-integrationclient.jar";"%IC_HOME%\log";
java.exe
-Dcom.taleo.integration.client.install.dir=%IC_HOME%
-Djava.endorsed.dirs=%IC_HOME%\lib\endorsed
-Djavax.xml.parsers.SAXParserFactory=org.apache.xerces.jaxp.SAXParserFactoryImpl
-Djavax.xml.transform.TransformerFactory=net.sf.saxon.TransformerFactoryImpl
-Dorg.apache.commons.logging.Log=org.apache.commons.logging.impl.Log4JLogger
-classpath %CLASSPATH% com.acme.integration.ACMEIntegrationProcess
```

The `IC_HOME` variable defines the installation directory of the Taleo Connect Client. This information is required because several libraries and resources are delivered with the installation bundle.

The classpath of the application is fairly simple because the `taleo-integrationclient.jar` contains the full classpath embedded in its manifest. This works fine with most JRE implementations. If any classpath problems occur, you may also explicitly define the classpath by including all Java libraries (*.jar) in the `lib` and `lib/endorsed` folders. Of course, the classpath must also contain the required references to the host Java application.

In the Java invocation command itself, we first set the Taleo Connect Client installation directory explicitly. This is required mainly to locate the feature pack files. The next two instructions ensure that the proper version of the Xerces XML parser is used. Then, because most of the XSLT files used in the Taleo Connect Client are in version 2.0 of this specification, we ensure that the XSL transformer supports it (in this case by choosing the Saxon 8 Basic engine). The final system property directs the Apache Jakarta Commons Logging project to use the Apache Log4J system. This is the reason why the `log` folder is part of the classpath, as the `log4j.properties` configuration file is located there. The invoked class `com.acme.integration.ACMEIntegrationProcess` is used here only as an example and should be replaced by the actual entry point of your Java application hosting the Taleo Connect Client.

The correct entry point to execute the Taleo Connect Client is: `com.taleo.integration.client.Client.execute()`. The JavaDoc provided for this method in the `docs/api` folder of the product describes the behavior and parameter values.
Starting

The Objective

The objective is to import one record into a Taleo product and then to export basic information from that same record back to TCC.

A new candidate will be imported into the Taleo Professional product. This example will create a candidate profile only, e.g. it will not create an application on a requisition.

Basic information regarding the candidate previously uploaded will be extracted. Various features of the export editor will be used to build powerful custom queries to extract data from the Taleo product.

The candidate has:

- An e-mail address
  psmith@acme.com
- A first name
  Paul
- A last name
  Smith
- A curriculum vitae provided as a Microsoft Word document
  psmith_cv.doc
- A few work experiences
  Senior Consultant at ABC International in 2007
  Analyst at XYZ Inc. for the first half of 2008

All the relevant files will be created in the C:\tcc\starter folder.

Creating Starter Folder

Steps

1. From Windows desktop open My Computer.
2. Open Local Disk (C:).
3. File > New > Folder.
   a) Rename New Folder tcc.
   b) Double click tcc
   Tcc is open, the address bar shows C:\tcc.
   a) Rename New Folder starter.
   b) Double click starter.
   starter is open, the address bar shows C:\tcc\starter.
5. Navigate to the Taleo Connect Client installation folder.
   a) Open monitor folder.
b) Copy web folder.
c) Navigate back to the starter folder.
d) Paste web folder

The HTML monitoring page needs the graphic elements from that folder to display properly.


Creating Curriculum Vitae

Steps
1. Open Microsoft Word or a compatible editor.
2. Create a new document.
3. Enter Curriculum Vitae Paul Smith.
4. Save as psmith_cv.doc in C:\tcc\starter.
5. Close editor.

Starting Taleo Connect Client

Steps
1. Start Taleo Connect Client.
   First the splash screen appears, then the Ping the Taleo Product window opens.
2. Select the product to connect to from the pull down menu.
   The product to ping is selected.
3. Select the Endpoint.
   Make sure the Endpoint selected is for a staging, development, or testing zone, not the production zone. The information in this procedure will truly be imported to your Taleo Professional product and there is no undo action possible.
4. Verify the authentication information.
5. Click Ping.
   The Ping the Taleo product window closes. The TCC graphic user interface opens.
Importing

Creating Import Specification

The first task is to create an import specification that will describe the structure of the CSV format request file, by defining the information represented by each column.

Steps
1. **File>New>New Import Wizard**
   
The New Import Wizard window opens
2. Select Professional/Recruiting [Version] product from pull down menu.
3. Select Candidate Import Service from list.
4. Select Create new import specification.
5. Select Merge Operation from pull down menu.
   
The merge operation will delegate processing to the create or the update operation as needed. If the record does not exist it will create it, if it exists it will update it.
6. Click Finish.
   
The New import specification is displayed in the import editor.

Defining Import Specification

Steps
1. Verify General tab information and correct if necessary.
2. Click on Entity tab.
   
The Entity view displays the Candidate entity structure.
3. Click on Columns tab.
   
The Columns tab displays an empty columns list.
4. Drag and drop EmailAddress from the Entity Structure to the Columns list.
   
a) In Behavior section check Use the value to search or lookup.
   
b) Verify that Target is Candidate.
   
c) Check Also set this value in the entity.
5. Drag and drop FirstName from the Entity Structure to the Columns list.
6. Drag and drop LastName from the Entity Structure to the Columns list.
7. Scroll Entity Structure to AttachedFiles and expand.
   
a) Expand CandidateAttachedFile.
   
b) Drag and drop Name from the Entity Structure to the Columns list.
   
c) Drag and drop Content from the Entity Structure to the Columns list.
8. Click AttachedFiles,Content in Columns list.
   
a) Select Value is the location of a file to attach.
   
Indicates that the field is the path and file name of the file to attach.
   In terms of the business model, the LastGeneralApplication relation is where we create a
   general profile for the candidate.

10. Drag and drop ProfileLocale from the Entity Structure to the Columns list.
    The language of the profile is a mandatory field.
    a) Click LastGeneralApplication,ProfileLocale in Columns list.
    b) Select Header check box in Column information section.
    c) Enter ProfileLocale as header label.

11. Scroll LastGeneralApplication to Experiences and expand.
    a) Expand Experience.

12. Drag and drop DisplaySequence from the Entity Structure to the Columns list.
    The display sequence of the experience is a mandatory field.

13. Click LastGeneralApplication,Experiences,DisplaySequence in Columns list.
    a) Select Header check box in Column information section.
    b) Enter Exp1-Sequence as header label.

14. Drag and drop OtherEmployerName from the Entity Structure to the Columns list.
15. Click LastGeneralApplication,Experiences,OtherEmployerName in Columns list.
    a) Select Header check box in Column information section.
    b) Enter Exp1-Employer as header label.

16. Drag and drop OtherJobTitle from the Entity Structure to the Columns list.
17. Click LastGeneralApplication,Experiences,OtherJobTitle in Columns list.
    a) Select Header check box in Column information section.
    b) Enter Exp1-JobTitle as header label.

18. Drag and drop BeginDate from the Entity Structure to the Columns list.
19. Click LastGeneralApplication,Experiences,BeginDate in Columns list.
    a) Select Header check box in Column information section.
    b) Enter Exp1-BeginDate as header label.

20. Drag and drop EndDate from the Entity Structure to the Columns list.
21. Click LastGeneralApplication,Experiences,EndDate in Columns list.
    a) Select Header check box in Column information section.
    b) Enter Exp1-EndDate as header label.

22. Drag and drop DisplaySequence from the Entity Structure to the Columns list.
    This is the beginning of the columns that describe the second work experience
23. Click LastGeneralApplication,Experiences,DisplaySequence in Columns list.
    a) Select Header check box in Column information section.
    b) Enter Exp2-Sequence as header label.
    c) Select Start a new element in the collection in Behavior section.
       Indicates the start of a new element in the experience collection.
    d) Select Skip column if value is empty in Behavior section.
       Indicates that a second experience is optional. This makes the specification more flexible.

24. Drag and drop OtherEmployerName from the Entity Structure to the Columns list.
25. Click LastGeneralApplication, Experiences, OtherEmployerName in Columns list.
   a) Select Header check box in Column information section.
   b) Enter Exp2-Employer as header label.
   c) Select Skip column if value is empty in Behavior section.

26. Drag and drop OtherJobTitle from the Entity Structure to the Columns list.
27. Click LastGeneralApplication, Experiences, OtherJobTitle in Columns list.
   a) Select Header check box in Column information section.
   b) Enter Exp2-JobTitle as header label.
   c) Select Skip column if value is empty in Behavior section.

28. Drag and drop BeginDate from the Entity Structure to the Columns list.
29. Click LastGeneralApplication, Experiences, BeginDate in Columns list.
   a) Select Header check box in Column information section.
   b) Enter Exp2-BeginDate as header label.
   c) Select Skip column if value is empty in Behavior section.

30. Drag and drop EndDate from the Entity Structure to the Columns list.
31. Click LastGeneralApplication, Experiences, EndDate in Columns list.
   a) Select Header check box in Column information section.
   b) Enter Exp2-EndDate as header label.
   c) Select Skip column if value is empty in Behavior section.

Saving Import Specification

Steps
1. File > Save as...
2. Select C:\tcc\starter as save in folder.
   This folder was created earlier.
3. Enter candidate_merge as file name.
4. Select import (*_ld.xml) for save as type.
5. Click Save.

Creating Candidate CSV File

Steps
1. Click Generate a sample csv file
   A sample csv file with the import specification columns opens in the default editor.
2. Enter column data.
   a) EmailAddress: psmith@acme.com
   b) FirstName: Paul
   c) LastName: Smith
   d) AttachedFiles,Name: psmith_cv.doc
   e) AttachedFiles,Content: C:\tcc\starter\psmith_cv.doc
f) ProfileLocale: en

g) Exp1-Sequence: 1

h) Exp1-Employer: ABC International

i) Exp1-JobTitle: Senior Consultant

j) Exp1-BeginDate: 2007-01-02

k) Exp1-EndDate: 2007-12-31

l) Exp2-Sequence: 2

m) Exp2-Employer: XYZ Inc.

n) Exp2-JobTitle: Analyst

o) Exp2-BeginDate: 2008-01-02

p) Exp2-EndDate: 2008-06-30

3. **File>Save as...**

4. Select C:\tcc\starter as save in folder.

5. Enter candidate_merge as file name.

6. Select csv files (*.csv) for save as type.

7. Click Save.

### Creating Import Configuration

**Steps**

1. **File>New>New Configuration Wizard**

   The New Configuration Wizard window opens.

2. Select Based on an import specification.

   a) Click on browse.

   b) Select C:\tcc\starter.

   c) Select candidate_merge.ld.xml.

   d) Click Open.

3. Click Next.

4. Verify Endpoint.

   Make sure the Endpoint selected is for a staging, development, or testing zone, not the production zone. The information in this procedure will truly be imported to your Taleo Professional product and there is no undo action possible.

5. Click Finish.

   The New configuration specification is displayed in the configuration editor.

### Defining Import Configuration

**Steps**

1. Verify general tab information and correct if necessary.

2. In Request select Pre-defined value.

   a) In file, click browse.
b) Look in: C:\tcc\starter.
c) Select candidate_merge.csv.
d) Click Open.

3. In Response select Generated using an identifier.
   a) In folder click browse button.
   b) Select C:\tcc\starter.
   c) In Identifier, click button.

   The Choose Identifier window opens.
   d) Enter [FILE]_}[NOW].
   e) Click OK.

4. Click Monitoring tab.
   a) Verify Enable Monitoring is checked.
   b) In folder click browse button.
   c) Select C:\tcc\starter.
   d) Verify Reuse the workflow identifier as the file name is selected.
   e) Verify Create XML monitoring files is checked.
   f) Verify Create HTML monitoring files is checked.
   g) Verify Create text monitoring files is checked.

5. Click Send tab.
   a) Verify Endpoint.

   Make sure the Endpoint selected is for a staging, development, or testing zone, not the
   production zone. The information in this procedure will truly be imported to your Taleo
   Professional product and there is no undo action possible.

   The Preview section displays the URL of the endpoint.

6. Click Poll tab.
   a) Verify Use Endpoint from the list is selected.
   b) Verify Endpoint is same as previous.
   c) In Quality section, verify Retry is checked.
   d) Verify Interval (in seconds) is 10
   e) Verify Completion attempts is checked and set to 30.

7. Click Retrieve tab.
   a) Verify Use Endpoint from the list is selected.
   b) Verify Endpoint is same as previous.

### Saving Import Configuration

**Steps**

1. *File>Save*
2. Select C:\tcc\starter as save in folder.
3. Enter candidate_merge as file name.
4. Select Client configuration (*.cfg.xml) for save as type.
5. **Click Save.**

**Executing Import Configuration**

**Prerequisite**
Import configuration file candidate-merge_cfg.xml is open and selected.

**Steps**

1. *Window>* **Open perspective > Runtime**
2. Click **Execute the configuration**.
3. Maximise Monitoring perspective
   The workflow execution details are displayed.
4. Wait for execution to complete.
5. Open C:\tcc\starter\candidate_merge-<datetime>.csv result file in CSV editor.
   The result file opens.
Getting Started

Exporting

Creating Export Specification

Steps
1. **File>New>New Export Wizard**
   
   The New Export Wizard window opens
2. Select Professional/Recruiting [Version] product from pull down menu.
3. In Filter enter the letter c.
   
   The candidate entity moves to the top of the list.
4. Select Candidate.
5. Verify Create new export is selected.
6. Click Finish.
   
   The New export specification is displayed in the import editor.

Defining Export Specification

Steps
1. Click on Entity.
   
   The entity view displays the Candidate fields and relations.
2. Click General tab.
   a) Verify Locale is English.
   b) In Export mode select CSV-report from pull down menu.
   c) Check CSV header present.
   d) Check CSV value delimiter and enter comma (,).
   e) Check CSV quotation character and enter double quote (").
3. Click Projections tab.
4. Drag and drop EmailAddress from the Entity Structure to the Projections list.
5. Click Add
   a) Select Add a function projection.
   b) Select Concatenate from the Function pull down menu.
   c) Click OK.
6. In Alias enter FullName
   a) Drag and drop LastName from the Entity Structure to the Function parameters list.
   b) Select LastName and move to top with up arrow.
   c) Select second parameter.
   d) Select String from Data type pull down menu.
   e) Enter a comma followed by a space (, ).
   f) Select last parameter and delete with minus.
g) Drag and drop FirstName from the Entity Structure to the Function parameters list.

7. Click Filters tab.
   a) Drag and drop EmailAddress from the Entity Structure to the Filters list.
   b) Click Filter (Equal) in the Filters list.
   c) Verify first value is EmailAddress.
   d) Verify first Data type is Field.
   e) Verify Operator is Equals.
   f) Enter psmith@acme.com in second Value.
   g) Verify Second Data type is String.

Saving Export Specification

Steps
1. File>Save as...
2. Select C:\tcc\starter as save in folder.
   This folder was created earlier.
3. Enter candidate_export as file name.
4. Select Export (*.sq.xml) for save as type.
5. Click Save.

Creating Export Configuration

Steps
1. File>New>New Configuration Wizard
   The New Configuration Wizard window opens.
2. Select Based on an export specification.
   a) Click on browse ...
   b) Select C:\tcc\starter.
   c) Select candidate_export_sq.xml.
   d) Click Open.
3. Click Next.
4. Verify Endpoint.
   Make sure the Endpoint selected is same one used for the record import.
5. Click Finish.
   The New configuration specification is displayed in the configuration editor.

Defining Export Configuration

Steps
1. Verify general tab information and correct if necessary.
2. In Request select Pre-defined value.
   a) In file, click browse  
   b) Look in: C:\tcc\starter.
   c) Select candidate_export_sq.xml.
   d) Click Open.
3. In Response select Generated using an identifier.
   a) In folder click browse  
   b) Select C:\tcc\starter.
   c) In Identifier, click  
      The Choose Identifier window opens.
   d) Enter Candidate-[NOW].
   e) Click OK.
4. Click Monitoring tab.
   a) Verify Enable Monitoring is checked.
   b) In folder click browse  
   c) Select C:\tcc\starter.
   d) Verify Reuse the workflow identifier as the file name is selected.
   e) Verify Create XML monitoring files is checked.
   f) Verify Create HTML monitoring files is checked.
   g) Verify Create text monitoring files is checked.
5. Click Send tab.
   a) Verify Endpoint.
      Make sure the Endpoint selected is same one used for the record import.
      The Preview section displays the URL of the endpoint.
6. Click Poll tab.
   a) Verify Use Endpoint from the list is selected.
   b) Verify Endpoint is same as previous.
   c) In Quality section, verify Retry is checked.
   d) Verify Interval (in seconds) is 10
   e) Verify Completion attempts is checked and set to 30.
7. Click Retrieve tab.
   a) Verify Use Endpoint from the list is selected.
   b) Verify Endpoint is same as previous.

**Saving Export Configuration**

**Steps**

1. **File>Save**
2. Select C:\tcc\starter as save in folder.
3. Enter candidate_export as file name.
4. Select Client configuration (*.cfg.xml) for save as type.
5. Click Save.

Executing Export Configuration

Prerequisite
Export configuration file candidate_export_cfg.xml is open and selected.

Steps
1. *Window* > *Open perspective* > *Runtime*
2. Click Execute the configuration
3. Maximise Monitoring perspective
   The workflow execution details are displayed.
4. Wait for execution to complete.
5. Open C:\tcc\starter\candidate-<datetime>.csv result file in CSV editor.
   The result file opens and displays the extracted data.
Using TCC

- Hierarchical Entities ................................................................. 102
- Filters ...................................................................................... 117
- Miscellaneous ....................................................................... 119
- Templates ............................................................................... 123
Hierarchical Entities

Hierarchical entities are organized in a tree structure. They are composed of parents and children. The root entity has no parent and is the ancestor of all the other entities.

Types

There are two types of hierarchical entities. The rigid type such as Organization, Location and JobField in SmartOrg and the flexible type such as Department in Professional.

Rigid hierarchical entities require special attention in import operations. To create a rigid type entity you must specify its Parent. To delete a rigid type entity you must first delete all its children or move them to a new Parent.

Flexible type entities pretty much behave as regular entities in import operations. They can be created individually without a parent or structured in a hierarchy. To delete a flexible type entity its dependencies, if any, must be reassigned to another entity.

Model

We will use the Organization entity to illustrate various import operations on hierarchical entities. We will create an Organization, update its multilingual fields, reorganize it, and delete parts of it.

Here is the top of the organization:

To keep the model manageable we have only expanded the sales division. Here is the sales division:
Once the Organization has been created and updated, the sales division will be restructured. Here is the restructured sales division:
Fields
We will use the following fields and relations from the Organization entity to perform our operations:

- **Code**: Unique identifier that provides a meaningful key. Unilingual string.
- **CustomerID**: Customer identifier used to reference the organization in an external system for integration purposes. Unilingual string.
- **Name**: Name of the organization instance. Multilingual string.
- **Parent, Organisation.Code**: Unique identifier that provides a meaningful key. Unilingual string.
- **Industry.Number**: Associated industry number that is pre-loaded in the Taleo application. Integer.
- **Sequence**: Specified sorting order that overrides the default sorting order when viewing the entity in the Taleo application. Integer.

Finding Root Code
To find the root code we will extract it from the entity. The root code value will be the parent code of the first record of the hierarchical entity we will create.

**Prerequisite**
Taleo Connect Client is running and ready for use.
Taleo application is available for integration.

**Steps**
1. From TCC open New Export Wizard *File>New>New Export Wizard*.
   The New Export Wizard opens.
2. In Product pull down menu, Select SmartOrg [Version].
   a) In Filter, enter letter o.
   b) Click on Organization.
   c) Select Create new export.
   d) Click on Finish.
   A New Export editor opens.
3. In General tab, select CSV header present.
We will use default values for the CSV attributes.

4. Select Projections tab.
   a) Drag and Drop Code, Name, and Level from Entity view to Projections list.

5. Select Filters tab.
   a) Drag Level from Entity view to Filters list and select it.
   b) In the bottom Data type drop down menu select Integer.
   c) In value, Enter 0.
      Level 0 is the root level.

6. Save as OrgRoot-export_sq.xml in organization-export folder.
   Folder structure is described in Installation section.

   The New Configuration Wizard opens.

8. Select Based on an export specification.
   a) Click on browser button
      A Select folder window opens.
   b) Navigate to organization-export folder.
   c) Select OrgRoot-export_sq.xml.
   d) Click on Open.
   e) Click on Next.
   f) Select Endpoint.
   g) Click on Finish.
      A New Configuration editor opens.

9. In General tab, under Request, select Pre-defined value
   a) Click on File browser button
      A Select file window opens.
   b) Navigate to organization-export folder.
   c) Select OrgRoot-export_sq.xml.
   d) Click on Open.
   The file path and name are displayed in the dialog box.

10. In General tab, under Response, select Generated using an identifier
    a) Click on Folder browser button
        A Select folder window opens.
    b) Navigate to organization-export folder.
    c) Click on OK.
        The file path is displayed in the dialog box
    d) Click on Identifier browser button
        A Choose Identifier window opens.
    e) In the Template dialog box enter [FILE]-[NOW]
f) Click OK.

[FILE]-[NOW] is displayed in the dialog box.

11. Save as OrgRoot-Extract_cfg.xml in organization-export folder.
12. Click on Execute the configuration and wait for the workflow to complete the extraction.
14. Open CSV result file OrgRoot-export_sq-yyyyMMddTHHmmss.csv in a csv editor or notepad

**Result**
The root code, organization name, and level are displayed below the header.

<table>
<thead>
<tr>
<th>Code, Name, Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROOT-ORGANIZATION, [en]=New Millenium, 0</td>
</tr>
</tbody>
</table>

### Creating an Organization

**Prerequisite**
Taleo Connect Client is running and ready for use.
Taleo application is available for integration.

**Steps**

1. From TCC open New Import Wizard **File>New>New Import Wizard.**
   The New Import Wizard opens.
2. In Product, Select SmartOrg [Version] from pull down menu.
   a) Click on Organization.
   b) Select Use the following import template.
   c) In Template pull down menu, select Basic Organization create.
   d) Click on Finish.
   A New Import editor opens.
3. Verify that the information in the General tab is correct.
   a) Header present is selected.
   b) Value delimiter is comma.
   c) Quotation character is double quotes.
4. In Advanced: Other import options click browser button ...
   a) Integration Processor is selected.
   b) olf is specified in dialog box.
   c) In Additional import options, switch.system.maintenance parameter is set to always.
   d) Click OK.
5. Click on Columns tab.
6. Select each column and verify Column information and Behavior.
7. Save as organization-create_ld.xml in organization-create folder.
8. Click Generate a sample csv file.
   The sample csv file opens in the default editor.
9. Enter the following data in the csv file.
The first line is the file header. The second line is the first line of data. ROOT-CODE is the root code extracted previously from the Taleo product.

<table>
<thead>
<tr>
<th>Code</th>
<th>CustomerID</th>
<th>Name</th>
<th>ParentCode</th>
<th>IndustryNumber</th>
<th>DisplaySequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPCORP0</td>
<td>CORP-CONPROD</td>
<td>Corporate Headquarters</td>
<td>ROOT-CODE</td>
<td>12,1</td>
<td></td>
</tr>
<tr>
<td>CPRD0</td>
<td>RD-CONPROD</td>
<td>Research &amp; Development</td>
<td>CPCORP0</td>
<td>12,300</td>
<td></td>
</tr>
<tr>
<td>CPRHR0</td>
<td>HR-CONPROD</td>
<td>Human Resources</td>
<td>CPCORP0</td>
<td>12,500</td>
<td></td>
</tr>
<tr>
<td>CPRPROD0</td>
<td>PROD-CONPROD</td>
<td>Production</td>
<td>CPCORP0</td>
<td>12,400</td>
<td></td>
</tr>
<tr>
<td>CPRFIN0</td>
<td>FINANCE-CONPROD</td>
<td>Finance</td>
<td>CPCORP0</td>
<td>12,200</td>
<td></td>
</tr>
<tr>
<td>CPSAL0</td>
<td>SALES-CONPROD</td>
<td>Sales</td>
<td>CPCORP0</td>
<td>12,100</td>
<td></td>
</tr>
<tr>
<td>CANSA0</td>
<td>SALES-CAN</td>
<td>Canada</td>
<td>CPSAL0</td>
<td>12,130</td>
<td></td>
</tr>
<tr>
<td>USASAL0</td>
<td>SALES-USA, United States of America</td>
<td>CPSAL0</td>
<td>12,110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MEXSAL0</td>
<td>SALES-MEX</td>
<td>Mexico</td>
<td>CPSAL0</td>
<td>12,120</td>
<td></td>
</tr>
<tr>
<td>CANSA1</td>
<td>SALES-MAR</td>
<td>Maritimes</td>
<td>CANSA0</td>
<td>12,131</td>
<td></td>
</tr>
<tr>
<td>CANSA2</td>
<td>SALES-QC</td>
<td>Quebec</td>
<td>CANSA0</td>
<td>12,132</td>
<td></td>
</tr>
<tr>
<td>CANSA3</td>
<td>SALES-ON</td>
<td>Ontario</td>
<td>CANSA0</td>
<td>12,133</td>
<td></td>
</tr>
<tr>
<td>CANSA4</td>
<td>SALES-PRA</td>
<td>Prairies</td>
<td>CANSA0</td>
<td>12,134</td>
<td></td>
</tr>
<tr>
<td>CANSA5</td>
<td>SALES-BC</td>
<td>British Columbia</td>
<td>CANSA0</td>
<td>12,135</td>
<td></td>
</tr>
<tr>
<td>USASAL1</td>
<td>SALES-NE</td>
<td>New England</td>
<td>USASAL0</td>
<td>12,111</td>
<td></td>
</tr>
<tr>
<td>USASAL2</td>
<td>SALES-MA</td>
<td>Mid-Atlantic</td>
<td>USASAL0</td>
<td>12,112</td>
<td></td>
</tr>
<tr>
<td>USASAL3</td>
<td>SALES-SE</td>
<td>Southeast</td>
<td>USASAL0</td>
<td>12,113</td>
<td></td>
</tr>
<tr>
<td>USASAL4</td>
<td>SALES-CGL</td>
<td>Central Great Lakes</td>
<td>USASAL0</td>
<td>12,114</td>
<td></td>
</tr>
<tr>
<td>USASAL5</td>
<td>SALES-SGC</td>
<td>South / Gulf Coast</td>
<td>USASAL0</td>
<td>12,116</td>
<td></td>
</tr>
<tr>
<td>USASAL6</td>
<td>SALES-MW</td>
<td>Midwest</td>
<td>USASAL0</td>
<td>12,115</td>
<td></td>
</tr>
<tr>
<td>USASAL7</td>
<td>SALES-WM</td>
<td>Western Mountains</td>
<td>USASAL0</td>
<td>12,117</td>
<td></td>
</tr>
<tr>
<td>USASAL8</td>
<td>SALES-CA</td>
<td>California</td>
<td>USASAL0</td>
<td>12,118</td>
<td></td>
</tr>
<tr>
<td>MEXSAL0</td>
<td>SALES-BA</td>
<td>Baja California</td>
<td>MEXSAL0</td>
<td>12,126</td>
<td></td>
</tr>
<tr>
<td>MEXSAL1</td>
<td>SALES-NM</td>
<td>Northern Mexico</td>
<td>MEXSAL0</td>
<td>12,125</td>
<td></td>
</tr>
<tr>
<td>MEXSAL2</td>
<td>SALES-TB</td>
<td>The Bajio</td>
<td>MEXSAL0</td>
<td>12,124</td>
<td></td>
</tr>
<tr>
<td>MEXSAL3</td>
<td>SALES-CM</td>
<td>Central Mexico</td>
<td>MEXSAL0</td>
<td>12,121</td>
<td></td>
</tr>
<tr>
<td>MEXSAL4</td>
<td>SALES-PC</td>
<td>Pacific Coast</td>
<td>MEXSAL0</td>
<td>12,123</td>
<td></td>
</tr>
<tr>
<td>MEXSAL5</td>
<td>SALES-YP</td>
<td>Yucatan Peninsula</td>
<td>MEXSAL0</td>
<td>12,122</td>
<td></td>
</tr>
</tbody>
</table>

10. Save file as organization-create.csv in organization-create folder.

The New Configuration Wizard opens.
12. Select Based on an import specification.
   a) Click on browser button ... A Select folder window opens.
   b) Navigate to organization-create folder.
   c) Select organization-create.ld.xml.
   d) Click on Open.
   e) Click on Next.
   f) Select Endpoint.
   g) Click on Finish.

A New Configuration editor opens.
13. In General tab, under Request, select Pre-defined value
   a) Click on File browser button ... A Select file window opens.
   b) Navigate to organization-create folder.
   c) Select organization-create.csv.
d) Click on Open.

The file path and name are displayed in the dialog box

14. In General tab, under Response, select Generated using an identifier
   a) Click on Folder browser button ...

      A Select folder window opens.

   b) Navigate to organization-create folder.

   c) Click on OK.

      The file path is displayed in the dialog box

   d) Click on Identifier browser button ...

      A Choose Identifier window opens.

   e) In the Template dialog box enter [FILE]-[NOW]

   f) Click OK.

      [FILE]-[NOW] is displayed in the dialog box

15. Save as organization-create_cfg.xml in organization-create folder.

16. Click on Execute the configuration and wait for the workflow to complete the import.

17. Open organization-create folder.

18. Open CSV result file organization-create-yyyyMMddTHHmsss.csv in a csv editor or notepad

   The following results are displayed.

```
"Index","Identifier","Status","TransactionType","Result","Message"
"1","","success","organization.create","",""
"2","","success","organization.create","",""
"3","","success","organization.create","",""
...
"26","","success","organization.create","",""
"27","","success","organization.create","",""
"28","","success","organization.create","",""
```

### Updating an Organization

Updating organisation multi-lingual fields with the specific locale method.

**Prerequisite**

Taleo Connect Client is running and ready for use.

Taleo application is available for integration.

Taleo application has more than one active language.

**Steps**

1. From TCC open New Import Wizard File>New>New Import Wizard.

   The New Import Wizard opens.

2. In Product, Select SmartOrg [Version] from pull down menu.
   a) Click on Organization.

   b) Select Use the following import template.

   c) In Template pull down menu, select Basic Organization update.
d) Click on Finish.

A New Import editor opens.

3. Verify that the information in the General tab is correct.
   a) Header present is selected.
   b) Value delimiter is comma.
   c) Quotation character is double quotes.

4. In Advanced: Other import options click browser button.

   a) Integration Processor is selected.
   b) olf is specified in dialog box.
   c) In Additional import options, switch.system mantenence parameter is set to always.
   d) Click OK.

5. Click on Columns tab.


   a) In Column information select Header and enter Name (en) in dialog box.
   b) In Behavior select Skip column if value is empty.
   c) Expand Multi-lingual fields.
   d) In locale type select Specific locale from pull down menu.
   e) In locale select english from pull down menu.

7. Drag and drop Name from Entity view.

   a) Select Organisation: name.
   b) Click Up button to move it above DisplaySequence and below Name (en).
   a) In Column information select Header and enter Nom (fr) in dialog box.
   b) In Behavior select Skip column if value is empty.
   c) In Multi-lingual fields, Locale type select Specific locale from pull down menu.
   d) In Multi-lingual fields, Locale select french from pull down menu.

8. Drag and drop Name from Entity view.

   a) Select Organisation: name.
   b) Click Up button to move it above DisplaySequence and below Nom (fr).
   c) In Column information select Header and enter Nombre (es) in dialog box.
   d) In Behavior select Skip column if value is empty.
   e) In Multi-lingual fields, Locale type select Specific locale from pull down menu.
   f) In Multi-lingual fields, Locale select spanish from pull down menu.

9. Save as organization-update_ld.xml in organization-update folder.

10. Click Generate a sample csv file.

    The sample csv file opens in the default editor.

11. Enter the following data in the csv file.

    The first line is the file header. The second line is the first line of data.

    | Code       | CustomerID       | Name (en)     | Nom (fr)   | Nombre (es) | DisplaySequence |
    |------------|------------------|---------------|------------|-------------|----------------|
    | CPCORP0    | CORP-CONPROD     | Corporate Headquarters, Siège social, Domicilio social | 1         |
    | CPRD0      | RD-CONPROD       | Research & Development, Recherche et développement, Investigación y desarrollo, 300 |
    | CPHRO      | HR-CONPROD       | Human Ressources, Ressources humaines, Recursos humanas, 500 |
    | CPPRD0     | PROD-CONPROD     | Production, Production, Producción, 400 |
12. Save file as organization-update.csv in organization-update folder.


The New Configuration Wizard opens.

14. Select Based on an import specification.

a) Click on browser button. 

A Select folder window opens.

b) Navigate to organization-update folder.

c) Select organization-update_id.xml.

d) Click on Open.

e) Click on Next.

f) Select Endpoint.

g) Click on Finish.

A New Configuration editor opens.

15. In General tab, under Request, select Pre-defined value

a) Click on File browser button. 

A Select file window opens.

b) Navigate to organization-update folder.

c) Select organization-update.csv.

d) Click on Open.
The file path and name are displayed in the dialog box.

   a) Click on Folder browser button. 
      A Select folder window opens.
   b) Navigate to organization-update folder.
   c) Click on OK.
      The file path is displayed in the dialog box.
   d) Click on Identifier browser button. 
      A Choose Identifier window opens.
   e) In the Template dialog box enter [FILE]-[NOW].
   f) Click OK.
      [FILE]-[NOW] is displayed in the dialog box.
17. Save as organization-update_cfg.xml in organization-export folder.
18. Click on Execute the configuration and wait for the workflow to complete the import.
20. Open CSV result file organization-update-yyyyMMddTHHmms.csv in a csv editor or notepad.

The following results are displayed.

```
"Index","Identifier","Status","TransactionType","Result","Message"
"1","","success","organization.update","",""
"2","","success","organization.update","",""
"3","","success","organization.update","",""
... 
"26","","success","organization.update","",""
"27","","success","organization.update","",""
"28","","success","organization.update","",""
```

Modifying an Organization

**Prerequisite**

Taleo Connect Client is running and ready for use.

Taleo application is available for integration.

**Steps**

1. From TCC open organization-create_ld.xml in organization-create folder.
2. Click Generate a sample csv file.
   The sample csv file opens in the default editor.
3. Enter the following data in the csv file.
   The first line is the file header. The second line is the first line of data.

<table>
<thead>
<tr>
<th>Code</th>
<th>CustomerID</th>
<th>Name</th>
<th>ParentCode</th>
<th>IndustryNumber</th>
<th>DisplaySequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMSAL1</td>
<td>SALES-NA</td>
<td>North America</td>
<td>CPSAL0</td>
<td>12,101</td>
<td></td>
</tr>
<tr>
<td>NMSAL2</td>
<td>SALES-EME</td>
<td>Europe &amp; Middle East</td>
<td>CPSAL0</td>
<td>12,102</td>
<td></td>
</tr>
<tr>
<td>NMSAL3</td>
<td>SALES-ASIA</td>
<td>Asia</td>
<td>CPSAL0</td>
<td>12,103</td>
<td></td>
</tr>
</tbody>
</table>
4. Save file as reorganization-create.csv in organization-create folder.
5. From TCC open organization-create_cfg.xml in organization-create folder.
6. In General tab, under Workflow modify Workflow identifier to Reorganization-create-[NOW].
7. In General tab, under Request, select Pre-defined value
   a) Click on File browser button ...
      A Select file window opens.
   b) Navigate to organization-create folder.
   c) Select reorganization-create.csv.
   d) Click on Open.
      The file path and name are displayed in the dialog box
8. Save as reorganization-create_cfg.xml in organization-export folder.
9. Click on Execute the configuration and wait for the workflow to complete the import.
10. Open organization-create folder.
11. Open CSV result file reorganization-create-yyyyMMddTHHmmss.csv in a csv editor or notepad
    The following results are displayed.
    | Index | Identifier | Status | TransactionType | Result | Message |
    |-------|------------|--------|-----------------|--------|---------|
    | 1     |            | success| organization.create |       |         |
    | 2     |            | success| organization.create |       |         |
    | 3     |            | success| organization.create |       |         |

**Moving an Organization**

**Prerequisite**
Taleo Connect Client is running and ready for use.
Taleo application is available for integration.

**Steps**
1. From TCC open New Import Wizard File>New>New Import Wizard.
   The New Import Wizard opens.
2. In Product, Select SmartOrg [Version] from pull down menu.
   a) Click on Organization.
   b) Select Use the following import template.
   c) In Template pull down menu, select Basic Organization move.
   d) Click on Finish.
   A New Import editor opens.
3. Verify that the information in the General tab is correct.
   a) Header present is selected.
   b) Value delimiter is comma.
   c) Quotation character is double quotes.
4. In Advanced: Other import options click browser button ...
   a) Integration Processor is selected.
   b) olf is specified in dialog box.
c) In Additional import options, switch.system.maintenance parameter is set to always.
d) Click OK.
5. Click on Columns tab.
6. Select each column and verify Column information and Behavior.
   The move operation requires two parameters, the organization to move and the new parent
   organization. The parameters are selected from the Parameter pull down menu in the Entity
   view.
7. Save as organization-move_id.xml in organization-move folder.
8. Click Generate a sample csv file.
   The sample csv file opens in the default editor.
9. Enter the following data in the csv file.
   The first line is the file header. The second line is the first line of data.

   "OrganizationCode","NewParentCode"
   CANSAL0,NMSAL1
   USASAL0,NMSAL1
   MEXSAL0,NMSAL1
10. Save file as organization-move.csv in organization-move folder.
    The New Configuration Wizard opens.
12. Select Based on an import specification.
    a) Click on browser button
        
        A Select folder window opens.
    b) Navigate to organization-move folder.
    c) Select organization-move_id.xml.
    d) Click on Open.
    e) Click on Next.
    f) Select Endpoint.
    g) Click on Finish.
        A New Configuration editor opens.
13. In General tab, under Request, select Pre-defined value
    a) Click on File browser button
        
        A Select file window opens.
    b) Navigate to organization-move folder.
    c) Select organization-move.csv.
    d) Click on Open.
    The file path and name are displayed in the dialog box
14. In General tab, under Response, select Generated using an identifier
    a) Click on Folder browser button
        
        A Select folder window opens.
    b) Navigate to organization-move folder.
Using TCC

Hierarchical Entities

c) Click on OK.
   The file path is displayed in the dialog box
d) Click on Identifier browser button ...

   A Choose Identifier window opens.
e) In the Template dialog box enter [FILE]-[NOW]
f) Click OK.
   [FILE]-[NOW] is displayed in the dialog box
15. Save as organization-move_cfg.xml in organization-move folder.
16. Click on Execute the configuration and wait for the workflow to complete the import.
17. Open organization-move folder.
18. Open CSV result file organization-move-yyyyMMddTHHmss.csv in a csv editor or notepad

   The following results are displayed.

<table>
<thead>
<tr>
<th>Index</th>
<th>Identifier</th>
<th>Status</th>
<th>TransactionType</th>
<th>Result</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>success</td>
<td>organization.move</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>success</td>
<td>organization.move</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>success</td>
<td>organization.move</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Deleting an Organization

Prerequisite
Taleo Connect Client is running and ready for use.
Taleo application is available for integration.

Steps
1. From TCC open New Import Wizard File>New>New Import Wizard.
   The New Import Wizard opens.
2. In Product, Select SmartOrg [Version] from pull down menu.
   a) Click on Organization.
   b) Select Use the following import template.
   c) In Template pull down menu, select Basic Organization delete.
   d) Click on Finish.
   A New Import editor opens.
3. Verify that the information in the General tab is correct.
   a) Header present is selected.
   b) Value delimiter is comma.
   c) Quotation character is double quotes.
4. In Advanced: Other import options click browser button ...
   a) Integration Processor is selected.
   b) ofl is specified in dialog box.
   c) In Additional import options, switch.system.maintenance parameter is set to always.
   d) Click OK.
5. Click on Columns tab.
6. Select each column and verify Column information and Behavior.
   The delete operation offers two parameters, the Organization to delete and the
   Reassign organization for the orphans. If the entity to delete has children, they can be reassigned
   to a new parent. The parameters are selected from the Parameter pull down menu in the Entity
   view.
7. Select Reassign Organisation and click Remove.
8. Save as organization-delete_ld.xml in organization-delete folder.
9. Click Generate a sample csv file.
   The sample csv file opens in the default editor.
10. Enter the following data in the csv file.
    The first line is the file header. The second line is the first line of data.
    | OrganisationCode |
    |------------------|
    | MEXSAL1          |
    | MEXSAL2          |
    | MEXSAL3          |
    | MEXSAL4          |
    | MEXSAL5          |
    | MEXSAL6          |
    | MEXSAL0          |
   Note that the parent is the last entry. All the children will be deleted before the parent.
11. Save file as organization-delete.csv in organization-delete folder.
   The New Configuration Wizard opens.
13. Select Based on an import specification.
    a) Click on browser button
    A Select folder window opens.
    b) Navigate to organization-delete folder.
    c) Select organization-delete_ld.xml.
    d) Click on Open.
    e) Click on Next.
    f) Select Endpoint.
    g) Click on Finish.
    A New Configuration editor opens.
14. In General tab, under Request, select Pre-defined value
    a) Click on File browser button
    A Select file window opens.
    b) Navigate to organization-delete folder.
    c) Select organization-delete.csv.
    d) Click on Open.
    The file path and name are displayed in the dialog box.
15. In General tab, under Response, select Generated using an identifier
a) Click on Folder browser button

A Select folder window opens.
b) Navigate to organization-delete folder.
c) Click on OK.

The file path is displayed in the dialog box
d) Click on Identifier browser button

A Choose Identifier window opens.
e) In the Template dialog box enter [FILE]-[NOW]
f) Click OK.

[FILE]-[NOW] is displayed in the dialog box
16. Save as organization-delete_cfg.xml in organization-move folder.
17. Click on Execute the configuration and wait for the workflow to complete the import.
19. Open CSV result file organization-delete-yyyyMMddTHHmmss.csv in a csv editor or notepad

The following results are displayed.

<table>
<thead>
<tr>
<th>Index</th>
<th>Identifier</th>
<th>Status</th>
<th>TransactionType</th>
<th>Result</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>success</td>
<td>organization.delete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>success</td>
<td>organization.delete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>success</td>
<td>organization.delete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>success</td>
<td>organization.delete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>success</td>
<td>organization.delete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>success</td>
<td>organization.delete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>success</td>
<td>organization.delete</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Filters

Last Run Date

The Last Run Date (LRD) is used in export specifications to filter records based on the value of a date field. This feature facilitates the extraction of records created or modified since the last execution of the workflow.

The LRD is set after the extract is completed and only if there were no errors during the process. The Last Run Date that is saved will be the one captured at the beginning of the process.

Files

The date and time of a successful export workflow is stored as a *.lrd file in the \Taleo Connect Client\lastrundates folder. The LRD file name is based on the configuration file that creates and updates it. For example: Candidate_export_cfg.xml creates or updates Candidate_export.lrd.

Because the LRD value is associated to a specific configuration file, each export specification that uses LRD filtering must be paired with its own configuration file. For example: Candidate_export_sq.xml is matched exclusively to Candidate_export_cfg.xml. Using multiple LRD export specifications with a single configuration file will cause saved LRD conflicts.

Delay

By default, LRD filtering uses the current zone time minus 5 minutes to set the LRD filter value. The default LRD minute adjustment can be modified with the following system property: com.taleo.integration.client.lrd.filtering.minute.adjustment.zone=[minute value]

Setting Last Run Date

Setting Last Run Date will create the *.lrd file with an LRD value for first time use.

Prerequisite

The corresponding export specification file that uses the LRD filter exists.

Steps

1. Create a new configuration based on the corresponding export specification.
2. Save the configuration file.
3. Click on the Pre-processing tab.
4. Click on Prepare Last Run Date.
5. Select time and date source.
   - TEE 7.0.5 and higher : Use date from the zone (default).
   - TEE 7.0.4 and lower : Use date from current computer.
6. Click on the Change link.
   a) Select date and time.
   b) Click OK.

Last Run Date value : displays selected date and time.
Strip Non Numeric Characters

Non numeric characters can be removed quickly from a projection with the REGEXP_REPLACE function.

Phone numbers are commonly entered in multiple formats:

- (123)456-7890
- (123) 456-7890
- 123-456-7890
- 123 456-7890

Example:

This projection removes non numeric characters to output the results in the same format (1234567890) with the REGEXP_REPLACE function.

```
<quer:projection alias="HomePhone">
    <quer:customFunction name="REGEXP_REPLACE">
        <quer:field path="Candidate,HomePhone"/>
        <quer:string>\D</quer:string>
        <quer:string/>
    </quer:customFunction>
</quer:projection>
```

Regular expressions are a powerful tool that can also be used for other data transformations.

Filtering vs ProjectionFiltering

`<filterings>` and `<projectionsFilterings>` have different semantics. This is to disambiguate where the filter should be applied: the driving table or a joined table.

For example, we want to create a query for the following model: Requisition -> CieLocation. This is an N relation where there are many CieLocation for one Requisition.

We create a query for Requisition and add a projection for CieLocation. If we have a `<filtering>` on CieLocation, it filters out Requisition that doesn’t satisfy the filter. If we have a `<projectionfiltering>` on CieLocation, it filters out CieLocation that doesn’t satisfy the filter.

In other words, `<filtering>` is a filter on an inner join and `<projectionfiltering>` is a filter on an outer join.
Miscellaneous

Removing Element From Collection

Taleo Connect Client can remove a specific element from a collection. When updating a record, the user can remove a specific element from a collection before adding new ones.

Prerequisite

Taleo Connect Client is running and ready for use.

Taleo application is available for integration.

- This operation is only available for removable relations. Please refer to the data dictionary entity relation summary table for documentation about removable relations.

Steps

1. From TCC open New Import Wizard File>New>Import Wizard.

   The New Import Wizard opens.

2. In Product, Select SmartOrg [Version] from pull down menu.
   a) Click on User.
   b) Select Create a new import specification.
   c) In Operation pull down menu, update.
   d) Click on Finish.

   A New Import editor opens.

3. Verify that the information in the General tab is correct.
   a) Header present is selected.
   b) Value delimiter is comma.
   c) Quotation character is double quotes.

4. Click on Columns tab.

5. Select the user search key.
   a) Drag User.CorrespondenceEmail from entity structure to Column list.
   b) In Behavior section, select Use the value for search or lookup.

6. Add a column to remove a Group from the user account.
   a) Drag UserAccount,Groups,Description from entity structure to Column list.
   b) In Behavior section, select Use the value for search or lookup.
   a) In Behavior section, select Remove element from the collection.

   To remove multiple items from the collection, repeat step as many times as required. In Behavior section, select "Start a new element in the collection" for the second and following items.

Exporting Language Codes

Language codes can be exported in CSV-report format. The user can identify the language code associated to multilingual fields, allowing the exported string to be matched to the locale.
**Prerequisite**

Taleo Connect Client is running and ready for use.

Taleo application is available for integration.

**Steps**

1. From TCC open New Export Wizard **File>New>New Export Wizard**.

   The New Export Wizard opens.

2. In Product pull down menu, Select SmartOrg [Version].
   
   a) In Filter, enter letter o.
   
   b) Click on Organization.
   
   c) Select Create new export.
   
   d) Click on Finish.

   A New Export editor opens.

3. In General tab, Other information, Export mode, select CSV-report from pull down menu. header present.

4. In General tab, Other information, select CSV header present.

   We will use default values for the CSV attributes.

5. Select Projections tab.

6. Drag and Drop Name from Entity view to Projections list.
   
   a) Select Name in projection list.
   
   b) Select Alias and enter ParentName in dialog box.
   
   c) Click Field open window button.

   d) Select Show the value only in the default locale.
   
   e) Click OK

7. Drag and Drop Code from Entity view to Projections list.
   
   a) Select Code in projection list.
   
   b) Select Alias and enter ParentCode in dialog box.

8. Drag and Drop Children,Name from Entity view to Projections list.

   Expand Children and BaseComposite to expose Name.

   a) Select Children,Name in projection list.
   
   b) Select Alias and enter ChildName in dialog box.
   
   c) Click Field open window button.

   d) Select Show values in all locales.
   
   e) Click OK.

9. Click Add.

   a) Select Add a multilingual field projection.
   
   b) Click OK.
   
   c) Select multilingual field projection.
   
   d) Select Alias and enter ChildLocale in dialog box.
   
   e) Select Locale from Field pull down menu.
   
   f) Drag and drop the Children relation into the Relation dialog box.
10. Select Filters tab.
   a) Drag Code from Entity view to Filters list and select it.
      By default first Value is set to Code, Data type is set to Field, and Operator is set to Equals.
   b) Set second Value to CPCORP0.
   c) Select String from pull down menu for second Data type.

11. Save as OrgLocale-export_sq.xml in organization-export folder.
    Folder structure is described in Installation section.

    The New Configuration Wizard opens.

13. Select Based on an export specification.
    a) Click on browser button ...
       A Select folder window opens.
    b) Navigate to organization-export folder.
    c) Select OrgLocale-export_sq.xml.
    d) Click on Open.
    e) Click on Next.
    f) Select Endpoint.
    g) Click on Finish.
       A New Configuration editor opens.

14. In General tab, under Request, select Pre-defined value
    a) Click on File browser button ...
       A Select file window opens.
    b) Navigate to organization-export folder.
    c) Select OrgLocale-export_sq.xml.
    d) Click on Open.
       The file path and name are displayed in the dialog box

15. In General tab, under Response, select Generated using an identifier
    a) Click on Folder browser button ...
       A Select folder window opens.
    b) Navigate to organization-export folder.
    c) Click on OK.
       The file path is displayed in the dialog box
    d) Click on Identifier browser button ...
       A Choose Identifier window opens.
    e) In the Template dialog box enter OrgLocale-[NOW]
    f) Click OK.
       OrgLocale-[NOW] is displayed in the dialog box

16. Save as OrgLocale-Extract_cfg.xml in organization-export folder.
17. Click on Execute the configuration and wait for the workflow to complete the extraction.
18. Open organization-export folder.
19. Open CSV result file OrgLocale-yyyyMMddTHHmmss.csv in a csv editor or notepad

Result

<table>
<thead>
<tr>
<th>ParentName</th>
<th>Parentcode</th>
<th>ChildName</th>
<th>ChildLocale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corporate Headquarters</td>
<td>CPCORP0</td>
<td>Research &amp; Development</td>
<td>en</td>
</tr>
<tr>
<td>Corporate Headquarters</td>
<td>CPCORP0</td>
<td>Investigación y desarrollo</td>
<td>es</td>
</tr>
<tr>
<td>Corporate Headquarters</td>
<td>CPCORP0</td>
<td>Recherche et développement</td>
<td>fr</td>
</tr>
<tr>
<td>Corporate Headquarters</td>
<td>CPCORP0</td>
<td>Human Ressources</td>
<td>en</td>
</tr>
<tr>
<td>Corporate Headquarters</td>
<td>CPCORP0</td>
<td>Recursos humanas</td>
<td>es</td>
</tr>
<tr>
<td>Corporate Headquarters</td>
<td>CPCORP0</td>
<td>Ressources humaines</td>
<td>fr</td>
</tr>
<tr>
<td>Corporate Headquarters</td>
<td>CPCORP0</td>
<td>Production</td>
<td>en</td>
</tr>
<tr>
<td>Corporate Headquarters</td>
<td>CPCORP0</td>
<td>Producción</td>
<td>es</td>
</tr>
<tr>
<td>Corporate Headquarters</td>
<td>CPCORP0</td>
<td>Production</td>
<td>fr</td>
</tr>
<tr>
<td>Corporate Headquarters</td>
<td>CPCORP0</td>
<td>Finance</td>
<td>en</td>
</tr>
<tr>
<td>Corporate Headquarters</td>
<td>CPCORP0</td>
<td>Finanzas</td>
<td>es</td>
</tr>
<tr>
<td>Corporate Headquarters</td>
<td>CPCORP0</td>
<td>Finance</td>
<td>fr</td>
</tr>
<tr>
<td>Corporate Headquarters</td>
<td>CPCORP0</td>
<td>Sales</td>
<td>en</td>
</tr>
<tr>
<td>Corporate Headquarters</td>
<td>CPCORP0</td>
<td>Ventas</td>
<td>es</td>
</tr>
</tbody>
</table>

UDF with Standard Name Export

How to export a UDF that has the same name as a standard field using TCC without any XML modification.

When a UDF and a standard field share the same name, the standard field is exported. If the UDF is a selection, you get an error message when trying to export the description of the code.

The workaround for this problem is adding "CustomField:" in the field name.

For example, "Candidate,Prefix" is a standard field. To extract the UDF, you must use "Candidate,CustomField:Prefix".

The same is true for a selection UDF, "Candidate,CustomField:Prefix,Description".

Integration with RSOFFER module

When a candidate selection workflow includes the RSOFFER, there are various integration restrictions/limitations using TCC.

The following tasks can be done using TCC:

- TCC can export draft offers and import them back using Offer number as key. Example would be to update Salary, Pay Basis and Pay Frequency Basis.
- After few manual steps, TCC can export the "approval pending" data and import them back as "offer approved" based on the approval list that's pre-defined.

The following tasks cannot be done using TCC:

- Cannot create a new offer record.
- Cannot use CSW MOVE operation to move from a step and status to another with in RSOFFER module.
- Cannot request for offer approval using TCC. It has to be a manual process.
Templates

Best Practice Templates

Best practice templates are included with TCC to give customers the foundations on which to build standard integration points.

Net Change

Net Change is used whenever possible for imports to Taleo zone. It reduces the possibility of errors by detecting the proper operation to apply according to its reference database. Use of Net Change also reduces the number of transactions making it easier to respect the zone volume limits.

Custom Fields

Custom field place holders are included in the templates. Make sure to synchronize the custom fields to update the data dictionary with user defined fields found in the Taleo product. You will then be able to replace the place holders with the required custom fields in the various integration points.

Candidate Standard Net Change

The candidate standard net change synchronizes the candidate data with the Taleo product.

- **Identifier**
  Use the same value as the Username or Email. This value will come back in the results file and facilitate reconciliation of the results.

- **UserName**
  This field is mandatory and must be unique. It is used as the primary key targeting the Candidate entity.

- **Password**
  When no password is provided, a default password is generated.

- **Language**
  This field is mandatory.

- **Location**
  Geographic location where the candidate resides.
  Valid Location Code.
  Must exist in Taleo Location Table.

- **LocationInterest**
  Locations where the candidate has expressed an interest to get a job.
  Required to populate EEO / EE fields. Must contain code from Location table to represent US or Canada if EEO/EE fields are populated.
  A candidate can have multiple locations of interest. Multiple locations of interest can be imported by adding as many locations of interest columns as necessary in TCC to cover the maximum a candidate can have according to your data.

- **CA/USA EE Question**
Remove country fields that do not apply.
Valid Question Number.
Must exist in Taleo Question Table.
• **CA/USA EE Answer**
  Remove country fields that do not apply.
  Valid Answer Number.
  Must exist in Taleo Answer Table.
• **ContestNumber-1**
  Search Target set to PreselectionApplication.
• **ContestNumber-1**
  This field is mandatory. The same information as ContestNumber-1 field above must be sent a second time with Net Change.
  Search Target set to Requisition.
• **ProfileStudyDisplaySequence-1**
  This field is mandatory when importing collections of studies.
• **ProfileStudyInstitution-1**
  Valid CielInstitution Number.
  Must exist in Taleo CielInstitution Table.
• **ProfileStudySubjectMajor-1**
  Valid CieProgram Description.
  Must exist in Taleo CieProgram Table.
• **ProfileStudyDegree-1**
  Valid StudyLevel Number.
  Must exist in Taleo StudyLevel Table.
• **ProfileExperienceDisplaySequence-1**
  This field is mandatory when importing collections of experiences.
• **CustomFieldName**
  Custom fields can be mandatory or optional depending on their settings.

**Candidate Standard Export**

The Candidate Standard Export uses the Candidate entity as the root entity to extract internal candidate information from the Taleo product.

**Projections**

All of the data is extracted as basic field projections.

**Filters**

• **Last Run Date**
  InclusiveLastModifiedDate is used as reference for last run date.
• **InternalApplication**
  The field value is compared to the value true for equality.
Department Standard Net Change

The department standard net change synchronizes the department data with the Taleo product.

- **Identifier**
  Use the same value as the Department number. This value will come back in the results file and facilitate reconciliation of the results.
- **DepartmentNumber**
  This field is mandatory and must be unique. It is used as the primary key targeting the Department entity.
- **DepartmentName-EN**
  This field is mandatory. Add one column for each language in which the Department Name will be imported. One column must be provided for each and every language supported by the application.
- **CostCenter**
  For customer reference; not used by Taleo. Add one column for each language in which the CostCenter will be imported. One column must be provided for each and every language supported by the application.
- **HiringManager**
  Valid Username for the Hiring Manager.
  Must exist in Taleo User Table.
- **CieLocation**
  Valid Location Code.
  Must exist in Taleo Location Table.
- **Organization**
  Valid Organization Code.
  Must exist in Taleo Organization Table.
- **ParentNumber**
  Valid Department number.
  Must exist in Taleo Department Table.
- **PayFactor**
  For customer reference; not used by Taleo
- **Recruiter**
  Valid Username for the Recruiter.
  Must exist in Taleo User Table.
- **State,Number**
  This field is mandatory.
- **DepartmentInformation,CustomFieldName**
  Custom fields can be mandatory or optional depending on their settings.

JobField Standard Net Change

The JobField standard net change synchronizes the JobField data with the Taleo product.
• **Identifier**
  Use the same value as the JobFieldCode. This value will come back in the results file and facilitate reconciliation of the results.

• **JobFieldCode**
  This field is mandatory and must be unique. It is used as the primary key targeting the JobField entity.

• **ParentCode**
  Valid JobFieldCode.
  Must exist in Taleo JobField Table.

• **Name-en**
  This field is mandatory. Add one column for each language in which the JobField Name will be imported. One column must be provided for each and every language supported by the application.

• **NetworkJobField**
  This field is mandatory and must be a valid NetworkJobField Number.

• **Status**
  This field is mandatory. 1-Active, 2-Draft (default), 3-Inactive, 4-Stored

• **Sequence**
  This field is mandatory. It sets the order of appearance of records in the application.

---

**JobTemplate Standard Net Change**

The JobTemplate standard net change synchronizes the JobTemplate data with the Taleo product.

• **Identifier**
  Use the same value as the JobCode. This value will come back in the results file and facilitate reconciliation of the results.

• **Creator**
  This field is mandatory and must be a valid User (Creator) Loginnname.

• **JobCode**
  This field is mandatory and must be unique. It is used as the primary key targeting the JobTemplate entity.

• **Title-EN**
  This field is mandatory. Add one column for each language in which the JobTemplate Title will be imported. One column must be provided for each and every language supported by the application.

• **Organization**
  Valid Organization Code.
  Must exist in Taleo Organization Table.

• **PrimaryLocation**
  Valid Location Code.
  Mandatory if EEO Job Category is populated.
  Must identify USA or Canada for EEO.

• **JobField**
  Valid JobField code.
• **EEOJobCategory**
  Used for either EEO Job Category OR client custom EEO Job Groups
  1=Officials & Managers; 2=Professionals; 3=Technicians; 4=Sales Workers; 5=Office/Clerical; 6=Craft Workers; 7=Operatives; 8=Laborers; 9=Service Workers.
  Other EEO Job Categories values can be configured and available if the EEO Job Group Number is enabled.

• **EmployeeStatus**
  1=Regular; 3=Contractual; 4=Temporary; 5=Limited Term.

• **JobLevel**
  1-Entry Level, 2-Individual Contributor, 3-Team leader, 4-Manager, 5-Director, 6-Executive, 7-Manager with direct Reports, 8-Manager without Direct reports 9-Senior Executive.

• **JobSchedule**
  1-Full-time, 2-Part-time, 3-Work-sharing, 4-Per Diem, 5-Contingent.

• **JobShift**
  1-Day Job, 2-Evening Job, 3-Night Job, 4-On Call, 5-Variable, 6-Flex Time, 7-Weekend, 8-Rotating.

• **JobType**
  1-Standard, 2-Internship, 3-Cooperative, 4-Summer Job, 5-Graduate Job, 6-Temporary Work, 7-Experienced.

• **OvertimeStatus**
  1-Exempt, 2-Non-exempt.

• **PayFrequencyBasis**
  1-Yearly, 2-Weekly, 3-Hourly, 4-Biweekly, 5-Semimonthly, 6-Monthly, 7-Total, 8-Daily.

• **StudyLevel**
  1-None, 2-High School Diploma/GED, 3-Technical Diploma, 4-Associate's Degree/College Diploma, 5-Non-Degree Program, 6-Bachelor's Degree, 7-Master's Degree, 8-Doctorate Degree, 9-Higher Degree, 10-Other.

• **WillTravel**
  1-No, 2-Yes, 25% of the Time, 3-Yes, 50% of the Time, 4-Yes, 75% of the Time, 5-Yes, 100% of the Time, 6-Yes, 5% of the Time, 7-Yes, 10% of the Time, 8-Yes, 15% of the Time, 9-Yes, 20% of the Time

• **DescriptionInternal-EN/DescriptionExternal-EN**
  1333 for multi-byte or 4000 for single byte. Add one column for each language in which the Descriptions will be imported. One column must be provided for each and every language supported by the application.

• **QualInternal-EN/QualExternal-EN**
  1333 for multi-byte or 4000 for single byte. Add one column for each language in which the Qualifications will be imported. One column must be provided for each and every language supported by the application.

• **UserGroup**
  Valid UserGroup CustomerCode.

• **RequisitionType**
  Possible values are 1=Professional (default); 2=Hourly; 3=Campus; 4= Contingent; 5=Executives

• **Status**
  Possible values are 1=Active (default); 2=Draft; 3=Inactive

• **CustomFieldName**
Custom fields can be mandatory or optional depending on their settings.

**Location Standard Net Change**

The Location standard net change synchronizes the Location data with the Taleo product.

- **Identifier**
  
  Use the same value as the LocationCode. This value will come back in the results file and facilitate reconciliation of the results.

- **LocationCode**
  
  This field is mandatory and must be unique. It is used as the primary key targeting the Location entity.

- **ParentCode**
  
  Valid Location Code.
  
  Must exist in Taleo Location Table.

- **Name-en**
  
  This field is mandatory. Add one column for each language in which the Location Name will be imported. One column must be provided for each and every language supported by the application.

- **NetworkLocation**
  
  This field is mandatory and must be a valid NetworkLocation Number.

- **Status**
  
  This field is mandatory. 1-Active, 2-Draft (default), 3-Inactive, 4-Stored

- **Sequence**
  
  This field is mandatory. It sets the order of appearance of records in the application.

**LUDS Standard Net Change**

The LUDS standard net change synchronizes the UDS Element data with the Taleo product.

- **Identifier**
  
  Use the same value as LUDSCode. This value will come back in the results file and facilitate reconciliation of the results.

- **LUDSCode**
  
  This field is mandatory and must be unique. It is used as the primary key targeting the UDSElement entity.

- **UDSCode**
  
  This field is mandatory and must be unique. It is used as the primary key targeting the UDSElement entity.

- **CustomerCode**
  
  This field is mandatory. It targets UserDefinedSelection entity.

- **Active**
  
  This field is mandatory.

- **Sequence**
  
  This field is mandatory for new records. Normally defaulted to the same value in all records, it is used to set order of appearance of records in the LUDS.
• **Description-EN**
  
  This field is mandatory for new records. Add one column for each language in which the Description will be imported. One column must be provided for each and every language supported by the application.

• **Organization**
  
  Valid Organization Code.
  
  Must exist in Taleo Organization Table.

• **Location**
  
  Valid Location Code.
  
  Must exist in Taleo Location Table.

• **JobField**
  
  Valid JobField Code.
  
  Must exist in Taleo JobField Table.

• **Complete**
  
  Optional field, it is strongly recommended to set the value to "true" otherwise the LUDS will appear as incomplete in the interface.

**Organization Standard Net Change**

The Organization standard net change synchronizes the Organization data with the Taleo product.

• **Identifier**
  
  Use the same value as the OrganizationCode. This value will come back in the results file and facilitate reconciliation of the results.

• **OrganizationCode**
  
  This field is mandatory and must be unique. It is used as the primary key targeting the Organization entity.

• **ParentCode**
  
  Valid OrganizationCode.
  
  Must exist in Taleo Organization Table.

• **Name-en**
  
  This field is mandatory. Add one column for each language in which the Organization Name will be imported. One column must be provided for each and every language supported by the application.

• **Industry**
  
  This field is mandatory and must be a valid Industry Number.

• **Status**
  
  This field is mandatory. 1-Active, 2-Draft (default), 3-Inactive, 4-Stored

• **Sequence**
  
  This field is mandatory. It sets the order of appearance of records in the application.

**Position Standard Net Change**

The Position standard net change synchronizes the Position data with the Taleo product.

• **Identifier**
Using TCC

Templates

Use the same value as the PositionID. This value will come back in the results file and facilitate reconciliation of the results.

- **DepartmentNumber**
  This field is mandatory and must be a valid DepartmentNumber.

- **HiringManager**
  This field is mandatory and must be a valid User Loginname.

- **JobCode**
  This field is mandatory and must be a valid JobCode.

- **PositionID**
  This field is mandatory and must be unique. It is used as the primary key targeting the Position entity.

**Requisition Standard Merge**

The requisition standard merge creates the requisition record if it does not exist or updates the record if it does exist.

- **Identifier**
  Use the same value as ContestNumber. This value will come back in the results file and facilitate reconciliation of the results.

- **ContestNumber**
  Requisition Number.
  This field is optional, if no ContestNumber is provided, one will be automatically assigned based on AWT settings.

- **HiringManager**
  Requisition hiring manager.
  This field is optional if JobStatus = 1 (Draft).
  This field is mandatory if JobStatus = 3 (Approved).

- **Owner**
  Requisition recruiter.
  This field is mandatory.

- **Language**
  This field is optional.

- **JobState**
  This field is optional, 1=Draft (default), 3=Approved.

- **CitizenshipStatus**
  1=Not Applicable, 2=Must Be a Citizen, 3=H1B Visa Accepted, 4=Other Visa Accepted.

- **EmployeeStatus**
  1=Regular, 3=Contractual, 4=Temporary, 5=Limited Term.

- **JobLevel**
  1-Entry Level, 2-Individual Contributor, 3-Team leader, 4-Manager, 5-Director, 6-Executive, 7-Manager with direct Reports, 8-Manager without Direct reports 9-Senior Executive.
1-Full-time, 2-Part-time, 3-Work-sharing, 4-Per Diem, 5-Contingent.

- **JobShift**
  1-Day Job, 2-Evening Job, 3-Night Job, 4-On Call, 5-Variable, 6-Flex Time, 7-Weekend, 8-Rotating.

- **JobType**
  1-Standard, 2-Internship, 3-Cooperative, 4-Summer Job, 5-Graduate Job, 6-Temporary Work, 7-Experienced.

- **Justification**
  1=New Position, 2=Replacement, 3=Modification, 4=Extension.

- **OvertimeStatus**
  1=Exempt, 2=Non-Exempt.

- **PayFrequencyBasis**
  1=yearly, 2=weekly, 3=hourly, 4=biweekly, 5=semimonthly, 6=monthly, 7=Total, 8=Daily.

- **StudyLevel**
  1=None, 2-High School Diploma/GED, 3-Technical Diploma, 4-Associate's Degree/College Diploma, 5-Non-Degree Program, 6-Bachelor's Degree, 7-Master's Degree, 8-Doctorate Degree, 9-Higher Degree, 10-Other.

- **StudyProgram**
  Valid CieProgram Description. Must exist in Taleo CieProgram Table.

- **WillTravel**
  1-No, 2-Yes, 25% of the Time, 3-Yes, 50% of the Time, 4-Yes, 75% of the Time, 5-Yes, 100% of the Time, 6-Yes, 5% of the Time, 7-Yes, 10% of the Time, 8-Yes, 15% of the Time, 9-Yes, 20% of the Time.

- **EEOJobCategory**
  Used for either EEO Job Category OR client custom EEO Job Groups 1=Officials & Managers; 2=Professionals; 3=Technicians; 4=Sales Workers; 5=Office/Clerical; 6=Craft Workers; 7=Operatives; 8=Laborers; 9=Service Workers- Other EEO Job Categories values can be configured and available if the EEO Job Group No is enabled.

- **JobField**
  This field is mandatory.
  Valid JobField code.

- **Organization**
  Valid Organization Code.
  Must exist in Taleo Organization Table.

- **PrimaryLocation**
  This field is mandatory.
  Valid Location Code.
  Must exist in Taleo Location Table.

- **UserGroup**
  This field is optional.
  Valid ArtGroup,CustomerCode.

- **Title-EN**
  Title of the requisition, this field is mandatory.
Add one field for each language in which the requisition Title is imported.

- **ReferenceTitle-EN**
  Manager WebTop Job Title, this field is mandatory.

Add one field for each language in which the requisition Reference Title is imported.

- **DescriptionInternal-EN/DescriptionExternal-EN**
  1333 multi-byte characters or 4000 single byte characters, maintain line feeds, carriage returns, and formatting embedded within field data.

Add one field for each language in which the internal or external description is imported.

- **QualInternal-EN/QualExternal-EN**
  1333 multi-byte characters or 4000 single byte characters, maintain line feeds, carriage returns, and formatting embedded within field data.

Add one field for each language in which the internal or external qualification is imported.

- **RequisitionType**
  This field is mandatory if many requisition types are activated in Taleo AWT. 1=Professional (default), 2=Hourly, 3=Campus, 4=Contingent, 5=Executives.

- **JobCode**
  This field is a link to JobTemplate, it can not be updated by integration.

- **CustomFieldName**
  Custom fields can be mandatory or optional depending on their settings.

### User Standard Net Change

The user standard net change synchronizes the user data with the Taleo product.

- **Identifier**
  Use the same value as the Loginname. This value will come back in the results file and facilitate reconciliation of the results.

- **Loginname**
  This field is mandatory and must be unique. It is used as the primary key targeting the User entity.

- **Password**
  This field is mandatory.

  Set Advanced options: updatable=false.

- **RolesCollection**
  When the collection is reset, all roles must be sent for each user. Also note that some business rules apply so that not all combinations of roles are possible.

- **Role-1**
  This field is mandatory.

- **Role-2**
  A user can have multiple roles.

  Multiple roles can be imported by adding as many role columns as necessary to cover the maximum a user can have according to your data.

- **FirstName**
  This field is mandatory.
• **LastName**
  This field is mandatory.

• **ContentLanguages**
  Languages in which the user will create requisitions and templates, en by default.
  Multiple languages are separated by a comma, for example: en,fr,es.
  These are the Career Section Languages.

• **CorrespondenceLanguage**
  Language used for correspondence with the user.
  These are the WebTop Languages.

• **ForceChangePassword**
  This field is mandatory.
  Set Advanced options: updatable=false.

• **Title-en**
  Add one field for each language in which the user Title is imported.

• **GroupCollection**
  When the collection is reset, all groups must be sent for each user.

• **Group-1**
  This field is mandatory.

• **Group-2**
  A user can belong to multiple groups.
  Multiple groups can be imported by adding as many group columns as necessary to cover the maximum a user can have according to your data.

• **CustomFieldName**
  Custom fields can be mandatory or optional depending on their settings.
Advanced Queries

- Introduction ..................................................................................................................... 135
- Inserting Complex Queries ............................................................................................ 136
- Examples ........................................................................................................................ 137
- Oracle Reserved Words ................................................................................................ 158
Introduction

In this section you will find examples of queries in SQL and their equivalent in XML. These examples can be used as models to create custom queries. Since the XML queries are converted to Oracle SQL queries during the extraction process, you will find a table of reserved words that can not be used as an alias in your XML queries.
Inserting Complex Queries

Prerequisite

Taleo Connect Client is running and ready for use.

An export editor based on the entity on which the complex queries will be applied is open.

This procedure applies to complex projections, complex filters, complex sorting, complex projection sorting, and all advanced tab functions.

Steps

1. Click on Projections tab.
   a) Click on Add.
   b) Select Add a complex projection.
   c) Click on OK.

   The Complex projection pane displays:

   ```xml
   ```

2. Enter or paste code in Complex projection pane.

Example:

```xml
  <quer:switchByValue>
    <quer:baseValue>
      <quer:field path="Candidate.Anonymous"/>
    </quer:baseValue>
    <quer:cases>
      <quer:case>
        <quer:integer>1</quer:integer>
        <quer:string>Anonymous candidate</quer:string>
      </quer:case>
      <quer:case>
        <quer:integer>0</quer:integer>
        <quer:string>Non anonymous candidate</quer:string>
      </quer:case>
    </quer:cases>
    <quer:defaultValue>
      <quer:string>Unknown</quer:string>
    </quer:defaultValue>
  </quer:switchByValue>
</quer:projection>
```
Examples

Between

SQL

SELECT CSUSER01.CSUSERNO AS CSUSERNO01
FROM CSUSER CSUSER01
WHERE (CSUSER01.CSUSERNO BETWEEN 1 AND 10))

XML query API

<filtering>
  <between
    <field path="no"/>
    <integer>1</integer>
    <integer>10</integer>
  </between>
</filtering>

Case (Switch by criterion)

SQL

SELECT CASE
  WHEN (CSUSER01.ANONYMOUS = 1) THEN
    'Anonymous candidate'
  WHEN (CSUSER01.ANONYMOUS = 0) THEN
    'Non anonymous candidate'
  ELSE
    'Unknown candidate'
END AS firstName01
FROM CSUSER CSUSER01

XML query API

<projection projectedField="firstName">
  <switchByCriterion>
    <cases>
      <case>
        <isTrue>
          <field path="anonymous"/>
        </isTrue>
        <string>Anonymous candidate</string>
      </case>
      <case>
        <isFalse>
          <field path="anonymous"/>
        </isFalse>
        <string>Non anonymous candidate</string>
      </case>
    </cases>
    <defaultValue>Unknown candidate</defaultValue>
  </switchByCriterion>
</projection>
Case (Switch by value)

SQL

```
SELECT CSUSER01.CSUSERNO AS CSUSERNO01,
CASE CSUSER01.ANONYMOUS
    WHEN 1 THEN
        'Anonymous candidate'
    WHEN 0 THEN
        'Non anonymous candidate'
    ELSE
        'Unknown candidate'
END AS firstName01
FROM CSUSER CSUSER01
```

XML query API

```
<projection projectedField="firstName">
    <switchByValue>
        <baseValue>
            <field path="anonymous"/>
        </baseValue>
        <cases>
            <case>
                <boolean>true</boolean>
                <string>Anonymous candidate</string>
            </case>
            <case>
                <boolean>false</boolean>
                <string>Non anonymous candidate</string>
            </case>
        </cases>
        <defaultValue>
            <string>Unknown candidate</string>
        </defaultValue>
    </switchByValue>
</projection>
```

Concatenate

SQL

```
SELECT ((CSUSER01.FIRSTNAME || ' ') || CSUSER01.LASTNAME) AS FIRSTNAME01
FROM CSUSER CSUSER01
```

XML query API

```
<query alias="testConcatenate" 
projectedClass="com.taleo.akirademo.entity.Candidate">
    <projections>
```

Connect By

SQL

SELECT STUDYLEVEL01.STUDYLEVELNO AS STUDYLEVELNO01
FROM STUDYLEVEL STUDYLEVEL01
WHERE 1 = 1
CONNECT BY PRIOR STUDYLEVEL01.STUDYLEVELNO = STUDYLEVEL01.PARENTSTUDYLEVELNO
START WITH (STUDYLEVEL01.STUDYLEVELNO = 5)

XML query API

<query alias="testHierarchicalFiltering"
    projectedClass="com.taleo.akirademo.entity.StudyLevel">
    <projections>
        <projection>
            <field path="no"/>
        </projection>
    </projections>
    <hierarchicalFilterings>
        <hierarchicalFiltering>
            <hierarchicalFilter>
                <equal>
                    <field path="no"/>
                    <field path="parent"/>
                </equal>
            </hierarchicalFilter>
            <startWithFilter>
                <equal>
                    <field path="no"/><integer>5</integer>
                </equal>
            </startWithFilter>
        </hierarchicalFiltering>
    </hierarchicalFilterings>
</query>

Count

SQL

SELECT COUNT((CSUSER01.CSUSERNO)) AS keyCount FROM CSUSER CSUSER01
XML query API

```
<query alias="testCount"
  projectedClass="com.taleo.akirademo.entity.Candidate">
  <projections>
    <projection alias="keyCount">
      <count>
        <field path="no"/>
      </count>
    </projection>
  </projections>
</query>
```

dbms_lob.getlength

**SQL**

```
SELECT dbms_lob.getlength(CSUSERATTACHEDFILE01.FILECONTENT) AS blobLength
FROM CSUSERATTACHEDFILE CSUSERATTACHEDFILE01
WHERE (CSUSERATTACHEDFILE01.CSUSERATTACHEDFILENO = 10291)
```

XML query API

```
<query alias="testLobLength"
  projectedClass="com.taleo.akirademo.entity.AttachedFile">
  <projections>
    <projection alias="blobLength">
      <lobLength>
        <field path="fileContent"/>
      </lobLength>
    </projection>
  </projections>
  <filterings>
    <filtering>
      <equal>
        <field path="no"/>
        <integer>10291</integer>
      </equal>
    </filtering>
  </filterings>
</query>
```

dbms_lob.instr

**SQL**

```
SELECT DBMS_LOB.INSTR(CSUSERATTACHEDFILE01.FILECONTENT, 'a') AS blobIndexOf
FROM CSUSERATTACHEDFILE CSUSERATTACHEDFILE01
WHERE (CSUSERATTACHEDFILE01.CSUSERATTACHEDFILENO = 10291)
```

XML query API

```
<query alias="testLobIndexOf"
  projectedClass="com.taleo.akirademo.entity.AttachedFile">
  <projections>
    <projection alias="blobIndexOf">
      <field path="fileContent"/>
    </projection>
  </projections>
  <filterings>
    <filtering>
      <equal>
        <field path="no"/>
        <integer>10291</integer>
      </equal>
    </filtering>
  </filterings>
</query>
```
<query alias="testLobIndexOf"
  projectedClass="com.taleo.akirademo.entity.AttachedFile">
  <projections>
    <projection alias="blobIndexOf">
      <lobIndexOf>
        <field path="fileContent"/>
        <string>205465737420</string>
      </lobIndexOf>
    </projection>
  </projections>
  <filterings>
    <filtering>
      <equal>
        <field path="no"/>
        <integer>10291</integer>
      </equal>
    </filtering>
  </filterings>
</query>

**dbms_lob.substr**

**SQL**

```sql
SELECT DBMS_LOB.SUBSTR(CUSERATTACHEDFILE01.FILECONTENT, 3, 4) AS blobSubstring
FROM CUSERATTACHEDFILE CUSERATTACHEDFILE01
WHERE (CUSERATTACHEDFILE01.CUSERATTACHEDFILENO = 10291)
```

**XML query API**

```xml
<query alias="testLobSubString"
  projectedClass="com.taleo.akirademo.entity.AttachedFile">
  <projections>
    <projection alias="blobSubString" projectedValueType="string">
      <lobSubString>
        <field path="fileContent"/>
        <integer>4</integer>
        <integer>3</integer>
      </lobSubString>
    </projection>
  </projections>
  <filterings>
    <filtering>
      <equal>
        <field path="no"/>
        <integer>10291</integer>
      </equal>
    </filtering>
  </filterings>
</query>
```

**Decode**

**SQL**

```sql
/* Decode */
```

---

User Guide  | Taleo Connect Client 12A  | 141
Advanced Queries

Examples

SELECT DECODE(CSUSER01.FIRSTNAME, 'Sebastien', 'Seb', 'Benoit', 'Ben', CSUSER01.FIRSTNAME) AS FIRSTNAME01
FROM CSUSER CSUSER01
WHERE 1 = 1

XML query API

<query alias="testDecode"
projectedClass="com.taleo.akirademo.entity.Candidate">
<projections>
    <projection>
        <decode>
            <field path="firstName"/>
            <string>Sebastien</string>
            <string>Seb</string>
            <string>Benoit</string>
            <string>Ben</string>
        </decode>
    </projection>
</projections>
</query>

Distinct

SQL

SELECT 1 FROM DUAL

XML query API

<query alias="MyDualQuery" projectedClass="DUAL_META_CLASS">
<projections>
    <projection alias="myProjection">
        <integer>1</integer>
    </projection>
</projections>
</query>

User Guide  Taleo Connect Client 12A  142
**Exists / Not Exists**

**SQL**

```
SELECT CSUSER01.FIRSTNAME AS FIRSTNAME01
FROM CSUSER CSUSER01
WHERE (EXISTS
  (SELECT CSUSER02.SALARY AS SALARY01 FROM CSUSER CSUSER02 WHERE 1 = 1))
```

```
SELECT CSUSER01.FIRSTNAME AS FIRSTNAME01
FROM CSUSER CSUSER01
WHERE NOT (EXISTS
  (SELECT CSUSER02.SALARY AS SALARY01 FROM CSUSER CSUSER02 WHERE 1 = 1))
```

**XML query API**

```
<query alias="testExists"
  projectedClass="com.taleo.akirademo.entity.Candidate">
  <projections>
    <projection>
      <field path="firstName"/>
    </projection>
  </projections>
  <filterings>
    <filtering>
      <returnsResults>
        <query alias="subQuery"
          projectedClass="com.taleo.akirademo.entity.Candidate">
          <projections>
            <projection>
              <field path="salary"/>
            </projection>
          </projections>
        </query>
      </returnsResults>
    </filtering>
  </filterings>
</query>
```

```
<query alias="testExists"
  projectedClass="com.taleo.akirademo.entity.Candidate">
  <projections>
    <projection>
      <field path="firstName"/>
    </projection>
  </projections>
  <filterings>
    <filtering>
      <not>
        <returnsResults>
          <query alias="subQuery"
            projectedClass="com.taleo.akirademo.entity.Candidate">
        </query>
      </not>
    </filtering>
  </filterings>
</query>
```
**Greater Than**

**SQL**

```
SELECT CSUSER01.CSUSERNO AS CSUSERNO01
FROM CSUSER CSUSER01
WHERE (CSUSER01.SALARY > 50)
```

**XML query API**

```
<query alias="testGreatherThan"
projectedClass="com.taleo.akirademo.entity.Candidate">
  <projections>
    <projection>
      <field path="no"/>
    </projection>
  </projections>
  <filterings>
    <filtering>
      <greaterThan>
        <field path="salary"/>
        <integer>50</integer>
      </greaterThan>
    </filtering>
  </filterings>
</query>
```

**Group By**

**SQL**

```
SELECT APPLICATION01.CANDIDATENO AS candidate,
       COUNT(APPLICATION01.CANDIDATENO) AS appCount
FROM APPLICATION APPLICATION01
WHERE 1 = 1
GROUP BY APPLICATION01.CANDIDATENO
```

**XML query API**

```
<query alias="testReportQueryWithGrouping"
projectedClass="com.taleo.akirademo.entity.BaseApplication">
```
<query alias="testGroupingFilteringInReportQuery" projectedClass="com.taleo.akirademo.entity.Candidate">
  <projections>
    <projection alias="city">
      <field path="city"/>
    </projection>
    <projection alias="cityCount">
      <count>
        <field path="city"/>
      </count>
    </projection>
  </projections>
  <groupings>
    <grouping>
      <field path="city"/>
    </grouping>
  </groupings>
  <groupingFilterings>
    <groupingFiltering>
      <greaterThan>
        <count>
          <field path="city"/>
        </count>
        <integer>5</integer>
      </greaterThan>
    </groupingFiltering>
  </groupingFilterings>
</query>

**Having**

**SQL**

```
SELECT CSUSER01.CITY AS CITY01, COUNT(CSUSER01.CITY) AS cityCount
FROM CSUSER CSUSER01
WHERE 1 = 1
GROUP BY CSUSER01.CITY
HAVING (COUNT(CSUSER01.CITY) > 5)
```
Advanced Queries

Examples

In

**SQL**

```sql
CSUSER01.CSUSERNO AS entityKey
FROM CSUSER CSUSER01
WHERE (CSUSER01.CSUSERNO IN (SELECT CSUSER02.CSUSERNO AS entityKey
FROM CSUSER CSUSER02
WHERE (CSUSER02.FIRSTNAME LIKE 's%' ESCAPE '\')))
```

```sql
SELECT CSUSER01.CSUSERNO AS entityKey
FROM CSUSER CSUSER01
WHERE (CSUSER01.CSUSERNO IN (1, 2, 3))
```

**XML query API**

```xml
<query alias="testIncludedIn" projectedClass="com.taleo.akirademo.entity.Candidate">
  <projections>
    <projection alias="entityKey" tag="entityKey">
      <key path=""/>
    </projection>
  </projections>
  <filterings>
    <filtering>
      <includedIn>
        <field path="no"/>
        <query alias="candidateSubQuery" projectedClass="com.taleo.akirademo.entity.Candidate">
          <projections>
            <projection alias="entityKey" tag="entityKey">
              <key path=""/>
            </projection>
          </projections>
          <filterings>
            <filtering>
              <matches>
                <field path="firstName"/>
                <string>s*</string>
              </matches>
            </filtering>
          </filterings>
        </query>
      </includedIn>
    </filtering>
  </filterings>
</query>
```

```xml
<query alias="testIncludedIn" projectedClass="com.taleo.akirademo.entity.Candidate">
  <projections>
    <projection alias="entityKey" tag="entityKey">
      <key path=""/>
    </projection>
  </projections>
</query>
```
Advanced Queries

Examples

Inner Join

**SQL**

```sql
SELECT APPLICATIONTRACKINGHISTORY01.APPLICATIONTRACKINGHISTORYNO AS APPLICATIONTRACKINGHISTORYNO01
FROM APPLICATIONTRACKINGHISTORY APPLICATIONTRACKINGHISTORY01,
APPLICATION APPLICATION01,
CSUSER CSUSER01
WHERE APPLICATION01.CANDIDATENO = APPLICATIONTRACKINGHISTORY01.CANDIDATENO
AND APPLICATION01.REQUISITIONNO = APPLICATIONTRACKINGHISTORY01.REQUISITIONNO
AND APPLICATION01.PROFILELOCALE = APPLICATIONTRACKINGHISTORY01.PROFILELOCALE
AND CSUSER01.CSUSERNO = APPLICATION01.CANDIDATENO
AND (CSUSER01.CSUSERNO > 0)
```

**XML query API**

```xml
<query alias="testInnerJoin"
projectedClass="com.taleo.akirademo.entity.ApplicationTrackingHistoryItem">
  <projections>
    <projection>
      <field path="no"/>
    </projection>
  </projections>
  <filterings>
    <filtering>
      <greaterThan><key path="application\,candidate="/>
        <integer>0</integer>
      </greaterThan>
    </filtering>
  </filterings>
</query>
```

Intersect

**SQL**

```sql
SELECT CSUSER01.CSUSERNO AS entityKey
```
FROM CSUSER CSUSER01
WHERE (CSUSER01.FIRSTNAME LIKE 'a%' ESCAPE '\')
INTERSECT
SELECT CSUSER02.CSUSERNO AS entityKey
FROM CSUSER CSUSER02
WHERE (CSUSER02.FIRSTNAME LIKE 's%' ESCAPE '\')

XML query API

<intersect alias="testIntersect">
  <queries>
    <query alias="testIntersect2" projectedClass="com.taleo.akirademo.entity.Candidate">
      <projections>
        <projection alias="entityKey" tag="entityKey">
          <key path=""/>
        </projection>
      </projections>
      <filterings>
        <filtering>
          <matches>
            <field path="firstName"/>
            <string>a*</string>
          </matches>
        </filtering>
      </filterings>
    </query>
    <query alias="candidateSubQuery" projectedClass="com.taleo.akirademo.entity.Candidate">
      <projections>
        <projection alias="entityKey" tag="entityKey">
          <key path=""/>
        </projection>
      </projections>
      <filterings>
        <filtering>
          <matches>
            <field path="firstName"/>
            <string>s*</string>
          </matches>
        </filtering>
      </filterings>
    </query>
  </queries>
</intersect>

Less Than

SQL

SELECT CSUSER01.CSUSERNO AS CSUSERNO01
FROM CSUSER CSUSER01
WHERE (CSUSER01.SALARY < 50)

XML query API
<query alias="testGreatherThan"
  projectedClass="com.taleo.akirademo.entity.Candidate">
  <projections>
    <projection>
      <field path="no"/>
    </projection>
  </projections>
  <filterings>
    <filtering>
      <lessThan>
        <field path="salary"/>
        <integer>50</integer>
      </lessThan>
    </filtering>
  </filterings>
</query>

**Like**

**SQL**

```sql
SELECT CSUSER01.CSUSERNO AS entityKey
  FROM CSUSER CSUSER01
  WHERE (CSUSER01.FIRSTNAME LIKE 's%' ESCAPE '\')
```

**XML query API**

```xml
<query alias="testLike"
  projectedClass="com.taleo.akirademo.entity.Candidate">
  <projections>
    <projection alias="entityKey" tag="entityKey">
      <key path=""/>
    </projection>
  </projections>
  <filterings>
    <filtering>
      <matches>
        <field path="firstName"/>
        <string>s*</string>
      </matches>
    </filtering>
  </filterings>
</query>
```

**Minus**

**SQL**

```sql
SELECT CSUSER01.CSUSERNO AS entityKey
  FROM CSUSER CSUSER01
  WHERE (CSUSER01.FIRSTNAME LIKE 'a%' ESCAPE '\')
MINUS
SELECT CSUSER02.CSUSERNO AS entityKey
  FROM CSUSER CSUSER02
  WHERE (CSUSER02.FIRSTNAME LIKE 's%' ESCAPE '\')
```
XML query API

<minus alias="testMinus">
  <queries>
    <query alias="candidate" projectedClass="com.taleo.akirademo.entity.Candidate">
      <projections>
        <projection alias="entityKey" tag="entityKey">
          <key path=""/>
        </projection>
      </projections>
      <filterings>
        <filtering>
          <matches>
            <field path="firstName"/>
            <string>a*</string>
          </matches>
        </filtering>
      </filterings>
    </query>
    <query alias="candidate2" projectedClass="com.taleo.akirademo.entity.Candidate">
      <projections>
        <projection alias="entityKey" tag="entityKey">
          <key path=""/>
        </projection>
      </projections>
      <filterings>
        <filtering>
          <matches>
            <field path="firstName"/>
            <string>s*</string>
          </matches>
        </filtering>
      </filterings>
    </query>
  </queries>
</minus>

NVL

SQL

SELECT NVL(CSUSER01.FIRSTNAME, 'Unknown') AS FIRSTNAME01
  FROM CSUSER CSUSER01
WHERE 1 = 1

XML query API

<query alias="testReplaceNull" projectedClass="com.taleo.akirademo.entity.Candidate">
  <projections>
    <projection>
      <replaceNull>
        <field path="firstName"/>
        <string>Unknown</string>
      </replaceNull>
    </projection>
  </projections>
</query>
Or

**SQL**

```sql
SELECT CSUSER01.CSUSERNO AS entityKey
FROM CSUSER CSUSER01
WHERE ((CSUSER01.FIRSTNAME LIKE 'a%' ESCAPE '\') OR
        (CSUSER01.FIRSTNAME = 'Sam'))
```

**XML query API**

```xml
<query alias="testIntersect2"
       projectedClass="com.taleo.akirademo.entity.Candidate">
  <projections>
    <projection alias="entityKey" tag="entityKey">
      <key path=""/>
    </projection>
  </projections>
  <filterings>
    <filtering>
      <or>
        <matches>
          <field path="firstName"/>
          <string>a*</string>
        </matches>
        <equal>
          <field path="firstName"/>
          <string>Sam</string>
        </equal>
      </or>
    </filtering>
  </filterings>
</query>
```

**Order By**

**SQL**

```sql
select csuserno from csuser order by csuserno asc
```

```sql
select csuserno from csuser order by csuserno desc
```

**XML query API**

```xml
<query alias="testOrderByAsc"
       projectedClass="com.taleo.akirademo.entity.Candidate">
  <projections>
    <projection>
      <field path="no"/>
    </projection>
  </projections>
</query>
```
Advanced Queries

Examples

Outer Join

SQL

SELECT APPLICATION01.CANDIDATENO AS CANDIDATENO01
FROM APPLICATIONTRACKINGHISTORY APPLICATIONTRACKINGHISTORY01,
APPLICATION APPLICATION01
WHERE APPLICATION01.CANDIDATENO(+) =
APPLICATIONTRACKINGHISTORY01.CANDIDATENO
AND APPLICATION01.REQUISITIONNO(+) =
APPLICATIONTRACKINGHISTORY01.REQUISITIONNO
AND APPLICATION01.PROFILELOCALE(+) =
APPLICATIONTRACKINGHISTORY01.PROFILELOCALE

XML query API

<query alias="testOuterJoin"
projectedClass="com.taleo.akirademo.entity.ApplicationTrackingHistoryItem">
  <projections>
    <projection>
      <field path="application+,candidate+/"/>
    </projection>
  </projections>
</query>

Rownum

SQL

SELECT CSUSER01.CSUSERNO AS CSUSERNO01, rowNum AS rowIndex
FROM CSUSER CSUSER01

User Guide  Taleo Connect Client 12A  152
WHERE 1 = 1

**XML query API**

```
<query alias="testRowindex"
    projectedClass="com.taleo.akirademo.entity.Candidate">
    <projections>
        <projection>
            <field path="no"/>
        </projection>
        <projection alias="rowIndex">
            <customValue>
                <expression>rowNum</expression>
            </customValue>
        </projection>
    </projections>
</query>
```

**SubString**

**SQL**

```
SELECT SUBSTR(CSUSER01.LASTNAME, 0, 2) AS LASTNAME01
    FROM CSUSER CSUSER01
    WHERE 1 = 1
```

**XML query API**

```
<query alias="testSubString"
    projectedClass="com.taleo.akirademo.entity.Candidate">
    <projections>
        <projection>
            <substring>
                <field path="lastName"/>
                <integer>0</integer>
                <integer>2</integer>
            </substring>
        </projection>
    </projections>
</query>
```

**ToChar**

**SQL**

```
SELECT TO_CHAR(LOCALTIMESTAMP(0)) AS tocharnow,
    TO_CHAR(LOCALTIMESTAMP(0), 'yyyy-MM-dd') AS tocharnowFormat,
    TO_CHAR(2.24567, '0.99') AS tochardouble
    FROM Dual Dual01
    WHERE 1 = 1
```

**XML query API**

```
<query alias="testToCharFunction" projectedClass="DUAL_META_CLASS">
```

WHERE 1 = 1
<projections>
    <projection alias="tocharnow">
        <toChar>
            <date type="now" xsi:nil="true"/>
        </toChar>
    </projection>
    <projection alias="tocharnowFormat">
        <toChar>
            <date type="now" xsi:nil="true"/>
            <string bound="false">yyyy-MM-dd</string>
        </toChar>
    </projection>
    <projection alias="tochardouble">
        <toChar>
            <double>2.7689340174</double>
            <string bound="false">0.99</string>
        </toChar>
    </projection>
</projections>
</query>

Trim

SQL

```
SELECT TRIM(CSUSER01.LASTNAME) AS LASTNAME01
FROM CSUSER CSUSER01
WHERE 1 = 1
```

XML query API

```
<query alias="testTrim"
    projectedClass="com.taleo.akirademo.entity.Candidate">
    <projections>
        <projection>
            <trim>
                <field path="lastName"/>
            </trim>
        </projection>
    </projections>
</query>
```

Truncate

SQL

```
SELECT TRUNC(CSUSER01.SALARY) AS SALARY01
FROM CSUSER CSUSER01
WHERE 1 = 1
```

XML query API

```
<query alias="testTruncate"
    projectedClass="com.taleo.akirademo.entity.Candidate">
    <projections>
        <projection>

```
**Advanced Queries**

**Examples**

```xml
<union alias="testUnion"/>
<queries>
  <query alias="testUnion2" projectedClass="com.taleo.akirademo.entity.Candidate">
    <projections>
      <projection alias="entityKey" tag="entityKey">
        <key path=""/>
      </projection>
    </projections>
    <filterings>
      <filtering>
        <matches>
          <field path="firstName"/>
          <string>a*</string>
        </matches>
      </filtering>
    </filterings>
  </query>
  <query alias="candidateSubQuery" projectedClass="com.taleo.akirademo.entity.Candidate">
    <projections>
      <projection alias="entityKey" tag="entityKey">
        <key path=""/>
      </projection>
    </projections>
    <filterings>
      <filtering>
        <matches>
          <field path="firstName"/>
          <string>s*</string>
        </matches>
      </filtering>
    </filterings>
  </query>
</queries>
```

**SQL**

```sql
SELECT CSUSER01.CSUSERNO AS entityKey
FROM CSUSER CSUSER01
WHERE (CSUSER01.FIRSTNAME LIKE 'a%' ESCAPE '\')
UNION
SELECT CSUSER02.CSUSERNO AS entityKey
FROM CSUSER CSUSER02
WHERE (CSUSER02.FIRSTNAME LIKE 's%' ESCAPE '\')
```

**XML query API**

```xml
<union alias="testUnion">
  <queries>
    <query alias="testUnion2" projectedClass="com.taleo.akirademo.entity.Candidate">
      <projections>
        <projection alias="entityKey" tag="entityKey">
          <key path=""/>
        </projection>
      </projections>
      <filterings>
        <filtering>
          <matches>
            <field path="firstName"/>
            <string>a*</string>
          </matches>
        </filtering>
      </filterings>
    </query>
    <query alias="candidateSubQuery" projectedClass="com.taleo.akirademo.entity.Candidate">
      <projections>
        <projection alias="entityKey" tag="entityKey">
          <key path=""/>
        </projection>
      </projections>
      <filterings>
        <filtering>
          <matches>
            <field path="firstName"/>
            <string>s*</string>
          </matches>
        </filtering>
      </filterings>
    </query>
  </queries>
</union>
```
Union All

SQL

```
SELECT CSUSER01.CSUSERNO AS entityKey
FROM CSUSER CSUSER01
WHERE (CSUSER01.FIRSTNAME LIKE 'a%' ESCAPE '\')
UNION ALL
SELECT CSUSER02.CSUSERNO AS entityKey
FROM CSUSER CSUSER02
WHERE (CSUSER02.FIRSTNAME LIKE 's%' ESCAPE '\')
```

XML query API

```
<unionAllowDuplicates alias="testUnionAll">
  <queries>
    <query alias="testUnion2" projectedClass="com.taleo.akirademo.entity.Candidate">
      <projections>
        <projection alias="entityKey" tag="entityKey">
          <key path=""></key>
        </projection>
      </projections>
      <filterings>
        <filtering>
          <matches>
            <field path="firstName"/>
            <string>a*</string>
          </matches>
        </filtering>
      </filterings>
    </query>
    <query alias="candidateSubQuery" projectedClass="com.taleo.akirademo.entity.Candidate">
      <projections>
        <projection alias="entityKey" tag="entityKey">
          <key path=""></key>
        </projection>
      </projections>
      <filterings>
        <filtering>
          <matches>
            <field path="firstName"/>
            <string>s*</string>
          </matches>
        </filtering>
      </filterings>
    </query>
  </queries>
</unionAllowDuplicates>
```

Other Custom Functions

SQL
SELECT CUSTOM_FUNCTION_NAME(myField,'param1','param2') AS myField01
FROM CSUSER CSUSER01
WHERE 1=1

**XML query API**

```xml
<query alias="testCustomFunction"
   projectedClass="com.taleo.akirademo.entity.Candidate">
  <projections>
    <projection>
      <customFunction name="CUSTOM_FUNCTION_NAME">
        <field path="myField"/>
        <string>param1</string>
        <string>param2</string>
      </customFunction>
    </projection>
  </projections>
</query>
```

**Custom Function Example**

**SQL**

```sql
SELECT REGEXP_REPLACE(CSUSER01.USERNAME,??) AS USERNAME01
FROM CSUSER CSUSER01
WHERE 1=1
```

**XML query API**

```xml
<query alias="testRegexpReplace"
   projectedClass="com.taleo.akirademo.entity.Candidate">
  <projections>
    <projection>
      <customFunction name="REGEXP_REPLACE">
        <field path="username"/>
        <string>a</string>
        <string>b</string>
      </customFunction>
    </projection>
  </projections>
</query>
```
# Oracle Reserved Words

Oracle reserved words can not be used as an Alias for an export query. The following table list words that are reserved in Oracle.

<table>
<thead>
<tr>
<th>Word</th>
<th>Word</th>
<th>Word</th>
<th>Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABORT</td>
<td>ACCEPT</td>
<td>ACCESS</td>
<td>ADD</td>
</tr>
<tr>
<td>ALL</td>
<td>ALTER</td>
<td>AND</td>
<td>ANY</td>
</tr>
<tr>
<td>ARRAY</td>
<td>ARRAYLEN</td>
<td>AS</td>
<td>ASC</td>
</tr>
<tr>
<td>ASSERT</td>
<td>ASSIGN</td>
<td>AT</td>
<td>AUDIT</td>
</tr>
<tr>
<td>AUTHORIZATION</td>
<td>AVG</td>
<td>BASE_TABLE</td>
<td>&gt;BEGIN</td>
</tr>
<tr>
<td>BETWEEN</td>
<td>BINARY_INTEGER</td>
<td>BODY</td>
<td>BOOLEAN</td>
</tr>
<tr>
<td>BY</td>
<td>CASE</td>
<td>CHAR</td>
<td>CHAR_BASE</td>
</tr>
<tr>
<td>CHECK</td>
<td>CLOSE</td>
<td>CLUSTER</td>
<td>CLUSTERS</td>
</tr>
<tr>
<td>COLAUTH</td>
<td>COLUMN</td>
<td>COMMENT</td>
<td>COMMIT</td>
</tr>
<tr>
<td>COMPRESS</td>
<td>CONNECT</td>
<td>CONSTANT</td>
<td>CRASH</td>
</tr>
<tr>
<td>CREATE</td>
<td>CURRENT</td>
<td>CURRVAL</td>
<td>CURSOR</td>
</tr>
<tr>
<td>DATA_BASE</td>
<td>DATABASE</td>
<td>DATE</td>
<td>DBA</td>
</tr>
<tr>
<td>DEBUGOFF</td>
<td>DEBUGON</td>
<td>DECIMAL</td>
<td>DECLARE</td>
</tr>
<tr>
<td>DEFAULT</td>
<td>DEFINITION</td>
<td>DELAY</td>
<td>DELETE</td>
</tr>
<tr>
<td>DESC</td>
<td>DIGITS</td>
<td>DISPOSE</td>
<td>DISTINCT</td>
</tr>
<tr>
<td>DO</td>
<td>DROP</td>
<td>ELSE</td>
<td>ELSIF</td>
</tr>
<tr>
<td>END</td>
<td>ENTRY</td>
<td>EXCEPTION</td>
<td>EXCEPTION_INIT</td>
</tr>
<tr>
<td>EXCLUSIVE</td>
<td>EXISTS</td>
<td>EXIT</td>
<td>FALSE</td>
</tr>
<tr>
<td>FETCH</td>
<td>FILE</td>
<td>FLOAT</td>
<td>FOR</td>
</tr>
<tr>
<td>FORM</td>
<td>FROM</td>
<td>FUNCTION</td>
<td>GENERIC</td>
</tr>
<tr>
<td>GOTO</td>
<td>GRANT</td>
<td>GROUP</td>
<td>HAVING</td>
</tr>
<tr>
<td>IDENTIFIED</td>
<td>IF</td>
<td>IMMEDIATE</td>
<td>IN</td>
</tr>
<tr>
<td>INCREMENT</td>
<td>INDEX</td>
<td>INDEXES</td>
<td>INDICATOR</td>
</tr>
<tr>
<td>INITIAL</td>
<td>INSERT</td>
<td>INTEGER</td>
<td>INTERFACE</td>
</tr>
<tr>
<td>Oracle Reserved Words</td>
<td>Oracle Reserved Words</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INTERSECT</td>
<td>INTO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LIKE</td>
<td>LIMITED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOOP</td>
<td>MAX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MINUS</td>
<td>MLSLABEL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MODIFY</td>
<td>NATURAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NEW</td>
<td>NEXTVAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOT</td>
<td>NOWAIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUMBER_BASE</td>
<td>OF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ONLINE</td>
<td>OPEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ORDER</td>
<td>OTHERS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PARTITION</td>
<td>PCTFREE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POSITIVEN</td>
<td>PRAGMA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRIVILEGES</td>
<td>PROCEDURE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RANGE</td>
<td>RAW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REF</td>
<td>RELEASE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESOURCE</td>
<td>RETURN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROLLBACK</td>
<td>ROW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROWNUM</td>
<td>ROWS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAVEPOINT</td>
<td>SCHEMA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SESSION</td>
<td>SET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIZE</td>
<td>SMALLINT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQLCODE</td>
<td>SQLERRM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STDDEV</td>
<td>SUBTYPE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNONYM</td>
<td>SYSDATE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TABLES</td>
<td>TASK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TO</td>
<td>TRIGGER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UID</td>
<td>UNION</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USE</td>
<td>USER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>LOCK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>MAXEXTENTS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>MODE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>NETWORK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>NOAUDIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>NOCOMPRESS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>NUMBER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>ON</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>OPTION</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>OR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>PACKAGE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>POSITIVE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>PRIVATE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>RAISE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>RECORD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>RENAME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>REVOKE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>ROWLABEL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>RUN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>SEPERATE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>SIGNTYPE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>SQL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>STATEMENT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>SUM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>TABLE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>THEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>TYPE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>UPDATE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>VALUES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VARCHAR</td>
<td>VARCHAR2</td>
<td>VARIANCE</td>
<td>VIEW</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>-------</td>
</tr>
<tr>
<td>VIEWS</td>
<td>WHEN</td>
<td>WHenever</td>
<td>WHERE</td>
</tr>
<tr>
<td>WHILE</td>
<td>WITH</td>
<td>WORK</td>
<td>WRITE</td>
</tr>
<tr>
<td>XOR</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
• Data Confidentiality ........................................................................................................ 162
• User Authentication ....................................................................................................... 163
• Stored Passwords ............................................................................................................ 164
• Specifying Integration Password .................................................................................. 165
• Override ENCRYPTED_PASSWORD ............................................................................. 166
• Encrypting Proxy Password .......................................................................................... 167
• HTTP Proxy .................................................................................................................... 168
• Encrypting Proxy Password .......................................................................................... 170
• Microsoft NTLM V1 Support ....................................................................................... 171
• Data at Rest ..................................................................................................................... 172
• TCC Plug-ins Security .................................................................................................... 173
Data Confidentiality

Taleo strongly recommends the enforcement of “Full SSL” as the supported communications protocol. The SSL communications protocol properties can be configured with the SmartOrg Communications Protocol Editor by a user with an Integrator role. When “Full SSL” mode is enforced, the Taleo-hosted environments only accept HTTPS requests. This communication protocol configuration ensures that only secure connections are accepted and that all data transmitted is properly encrypted. All Integration points created with TCC should conform to this protocol at the handshake.

To support organizations with limited HTTPS capabilities, TCC allows HTTP based communications. When this protocol is selected, the data is not encrypted during transmission between TCC and the Taleo-hosted environments. HTTP communication should only be used when the organization’s network does not support HTTPS communications.

IMPORTANT NOTE: All Taleo Enterprise Edition (TEE) product versions support SSL communications (HTTPS). However, only TEE, version 7.5 with Solution Pack 5 and later will exclusively enforce the "Full SSL" configuration for integration purposes. Previous versions will continue to support both HTTP and HTTPS integration requests, even if "Full SLL" is configured.
User Authentication

TCC supports both Web Services Security (WS-S) and Basic username/password authentication schemes.

Web Services Security (WS-S)
TCC supports Web Services Security Username Token Profile 1.1 using the PasswordDigest scheme. Using this authentication method, passwords are not sent over as part of the transaction. TCC uses the password to generate the password digest with the following algorithm:

PasswordDigest = Base64 (SHA-1 (nonce + creation timestamp + password))

TCC concatenates the nonce, the creation timestamp, and the password; generates the PasswordDigest with the SHA-1 hash algorithm; and Base64 encodes the result. This makes the password unintelligible and offers a basis for preventing replay attacks.

The password digest security is an effective defense against attacks directed at passwords themselves, such as Brute Force exploits. The passwords must be strong enough so that known attacks will not allow revealing secrets from a captured message.

Even if the password is not sent over the line, Taleo strongly recommends using SSL communications protocol (HTTPS).

- The WS-S authentication scheme is supported by the following Product versions:
  - Taleo Enterprise Edition (TEE) 7.5 SP6 and later
  - TCC 7.5 SP6 and later

Basic Username/Password
When using the Basic username/password authentication scheme, TCC provides user credentials in the form of a user name and password, as HTTP headers.

Unless the communications are secured using the HTTPS protocol, this information is sent in clear text. This is why Taleo strongly recommends using the HTTPS communications protocol.
 Stored Passwords

The Stored Passwords are used by TCC to establish communication with Taleo products and for authentication with proxy servers.

 Integration User password

TCC must maintain the Integration User password locally to establish a connection with a Taleo Product. The password can be stored in the following locations:

- Default configuration board file:
  default.board.xml TCC 7.5 SP6 and later
- TCC environment file:
  *.properties file TCC 7.5 SP5 and prior.
- TCC configuration files:
  *_cfg.xml

The Integration User password is encrypted by an AES algorithm with a key size of 128 bits. It is stored in an XML file, as follows: EncryptedPassword = Base64 (AES:128 (password)).

Password encryption using an AES 128-bit algorithm is supported by TCC 7.5 SP6. Previous versions are leveraging Taleo's proprietary algorithms to scramble the Integration User's password within the configuration files indicated above.

Taleo strongly recommends upgrading to TCC 7.5 SP6 to benefit from enhanced security features.

 Proxy password

The System properties are used by TCC to authenticate itself to proxy servers it will access. The required System properties are:

-Dhttp.proxyHost=myproxyhostname
-Dhttp.proxyPort=portnumber
-Dhttp.proxyUser=username
-Dhttp.proxyPassword=hello123

There are two different ways to set up System properties, depending on the way TCC is launched.

- Launching TCC with the Windows graphic user interface (GUI):
  When running TCC in the Windows GUI, all System properties must be defined in the TaleoConnectClient.ini file located at the root of the installation folder.
- Launching TCC in a Command Line interface:
  When running TCC in a Command Line interface, the System properties must be included in the TaleoConnectClient.bat (Windows) or TaleoConnectClient.sh (Unix/Linux) files.

With TCC 7.5 SP6 and later, proxy password encryption will be possible with a command line utility that uses AES 128 bit ciphers. The utility uses the password to encrypt as an argument to generate the new encrypted password. The encrypted password is inserted in the system properties as the proxy password. TCC will decrypt the password when needed.

Taleo strongly recommends upgrading to TCC 7.5 SP6 to benefit from enhanced security features.
Specifying Integration Password

Specifying Integration user password as ENCRYPTED_PASSWORD symbol when using the command line to invoke TCC.

Using ENCRYPTED_PASSWORD symbol improves security since the integration passwords do not have to be unveiled to TCC script developers.

Steps

1. Encrypt the integration password.
   a) Use the command line tool provided by TCC.

   ```
   [TCCHome]>EncryptPassword.bat TheSecretPassword
   Encrypting password using command line argument.
   Encrypted password:
   xHJD4X3xuVIMz7pOlfdxcFUBULFWGrZqTfNDw/DLpo=
   ```
   b) Copy the encrypted password to the clipboard.

2. Start TCC.

3. Open the default configuration board, **File>Open>Configuration Board**.

4. Create a new symbol.
   a) Name: ENCRYPTED_PASSWORD
   b) Value: `[DECRIPT("paste generated password here")]`

   Make sure to use the complete syntax using all the control characters: square brackets, parenthesis, and double-quotes.

5. Update the endpoint and use the `[ENCRYPTED_PASSWORD]` symbol in the Password field.

   Make sure the symbol is enclosed in square brackets as shown.

   **Note:** The runtime symbol function DECRYPT can only be used in Authentication Password fields.
Override ENCRYPTED_PASSWORD

To pass the password at the command line, add the following system property when invoking the TCC application:

- clear text password
  -Dcom.taleo.client.symbol.ENCRYPTED_PASSWORD=TheSecretPassword
- encrypted password
  -Dcom.taleo.client.symbol.ENCRYPTED_PASSWORD=[DECRYPT("xH/JD4X3xuVTMz7pO1fdxcPUBU1FWGrZqTfNDw/DLpo=")]
Encrypting Proxy Password

Prerequisite
Taleo Connect Client 7.5 Service Pack 6 or later.

Steps
1. Open a command line interface.
2. Execute the utility with the password as argument.
   
   ```
   C:\Program Files\Taleo Connect Client>EncryptPassword.bat
   MySecre7P4ssword
   ```
   
   Encrypting password using command line argument.
   Encrypted password:
   
   F0r6f9UebDtC+5cOeI0z0WQxYxA6m410vqqTr9OR1FE=

3. Copy encrypted password.
4. Open system properties file.
5. Paste encrypted password as `Dhttp.proxyPassword`.
   
   `-Dhttp.proxyPassword=F0r6f9UebDtC+5cOeI0z0WQxYxA6m410vqqTr9OR1FE=  

User Guide Taleo Connect Client 12A 167
HTTP Proxy

In certain corporate networks, IT policies enforce the use of a proxy to funnel all external HTTP requests. This is an important part of the overall network security.

The Taleo Connect Client fully supports the use of an HTTP proxy both in the runtime execution script and through the design environment. The configuration of the HTTP proxy follows the Java standard documented at http://java.sun.com/j2se/1.4.2/docs/guide/net/properties.html (these are the same specifications as for the new 1.5 version).

According to these specifications, the configuration of the HTTP proxy is done at the JRE level; that is the runtime environment in which the integration process is invoked. The connectivity information is passed as system properties in either the runtime execution script or the design environment initialization file.

Hence, assuming the corporate HTTP proxy is available at `proxy.acme.net` on port `3128`, the `TaleoConnectClient.bat` script should be changed to add:

```bash
%JAVA_HOME%\bin\java.exe
-Dhttp.proxyHost=proxy.acme.net
-Dhttp.proxyPort=3128
-Dhttp.proxyUser=acmeuser
-Dhttp.proxyPassword=secret
-Djava.endorsed.dirs=%IC_HOME%/lib/endorsed
```

Certain corporate HTTP proxy implementations require an explicit authentication. This information can be provided with two extra system properties as in the following example:

```bash
%JAVA_HOME%\bin\java.exe
-Dhttp.proxyHost=proxy.acme.net
-Dhttp.proxyPort=3128
```

Please note that we also fully support the `http.nonProxyHosts` property as defined in the Java specifications.

If the use of an HTTP proxy is required also in the design environment, then the following lines should be added to the `TaleoConnectClient.ini` file:

```ini
vmargs
-Djava.endorsed.dirs=lib/endorsed
-Dorg.apache.commons.logging.Log=org.apache.commons.logging.impl.Log4jLogger
-Djavax.xml.transform.TransformerFactory=net.sf.saxon.TransformerFactoryImpl
-Dhttp.proxyHost=proxy.acme.net
-Dhttp.proxyPort=3128
```

Certain corporate HTTP proxy implementations require an explicit authentication. This information can be provided with two extra system properties as in the following example:

```bash
%JAVA_HOME%\bin\java.exe
-Dhttp.proxyHost=proxy.acme.net
-Dhttp.proxyPort=3128
-Dhttp.proxyUser=acmeuser
-Dhttp.proxyPassword=secret
-Djava.endorsed.dirs=%IC_HOME%/lib/endorsed
```

Please note that we also fully support the `http.nonProxyHosts` property as defined in the Java specifications.

If the use of an HTTP proxy is required also in the design environment, then the following lines should be added to the `TaleoConnectClient.ini` file:

```ini
vmargs
-Djava.endorsed.dirs=lib/endorsed
-Dorg.apache.commons.logging.Log=org.apache.commons.logging.impl.Log4JLogger
-Djavax.xml.transform.TransformerFactory=net.sf.saxon.TransformerFactoryImpl
-Dhttp.proxyHost=proxy.acme.net
-Dhttp.proxyPort=3128
```
In previous versions, the configuration of the HTTP proxy was done in the client configuration file itself. This is no longer supported in favor of the standard way described above (see the section for more details).

Certain implementations support the `https.proxyHost` property although it is not part of the Java standard. The Taleo Connect Client only supports `http.proxyHost`, which is used for all HTTP requests (HTTP or HTTPS protocol).
Encrypting Proxy Password

Prerequisite
Taleo Connect Client 7.5 Service Pack 6 or later.

Steps
1. Open a command line interface.
2. Execute the utility with the password as argument.
   
   C:\Program Files\Taleo Connect Client>EncryptPassword.bat
   
   MySecre7P4ssword
   
   Encrypting password using command line argument.
   
   Encrypted password:
   
   F0r6f9UebDtC+5cOeI0z0WQxYxA6m410vqqTr9OR1FE=

3. Copy encrypted password.
4. Open system properties file.
5. Paste encrypted password as Dhttp.proxyPassword.

- Dhttp.proxyPassword=F0r6f9UebDtC+5cOeI0z0WQxYxA6m410vqqTr9OR1FE=
Microsoft NTLM V1 Support

The Microsoft NTLM V1 proxy server authentication scheme is now supported. Clients with network security policies leveraging only this authentication method can now install TCC within their infrastructure in compliance with their internal policy.

Things to Consider

- Technical limitations prevents password digest retrieval from the system.
- The proxy password cannot be read from the network automatically; it must be manually entered in the batch file.
- Users must keep the password in the batch file synchronized with the Windows domain password.
- All proxy passwords MUST be encrypted.

Proxy System Properties Configuration

The Microsoft NTLM V1 proxy server authentication scheme must be configured in files located at the root of the TCC installation folder.

- **Launching TCC using the Windows user interface**
  - When running TCC on Windows, all system properties must be defined in the TaleoConnectClient.ini file.
- **Launching TCC using the command line**
  - When running TCC using the command line, all system properties must be passed using the command line. Insert the system properties in the TaleoConnectClient.bat file for Windows or in the TaleoConnectClient.sh file for Unix/Linux.

System Properties for NTLM Proxies

The following list contains system properties you can configure for the proxy:

- `-Dhttp.proxyHost=myproxyhostname`
- `-Dhttp.proxyPort=8080`
- `-Dhttp.proxyUser=username` (Windows Domain User Name)
- `-Dhttp.proxyPassword=encrypted password` (Windows Domain Password)
- `-Dhttp.proxyNTDomain=ACMEDOMAIN` (Windows Domain Name)

**Note:** "http.proxyNTDomain" is only required for NTML proxy authentication.

Windows Vista

Although Microsoft Vista supports both the NTLM V1 and NTLM V2 proxy server authentication scheme, TCC currently supports only Microsoft NTLM V1.
Data at Rest

Standard TCC implementations have no data retention capabilities. Data-at-rest encryption capabilities are not required.
TCC Plug-ins Security

TCC functionalities can be extended using optional plug-ins. Please refer to plug-ins documentation for related security issues.
• Appendix A.....................................................................................................................175
• Appendix B.....................................................................................................................178
• Appendix C.................................................................................................................... 180
Appendix A

Volume Limits

The volume limits specify maximums for the number of records, allowed time, and MB of data per transaction or day. It also specifies the number of concurrent requests by product.

Purpose

The need for integration limits arises from the hosted service model of Taleo applications. All users of Taleo applications share a number of resources including web servers, application servers, and database servers. The actions of one user affect the overall resource availability.

The integration limits ensure that all users have a fair and reasonable access to server resources. They also ensure that the overall performance of your career sites and of your staffing applications cannot be affected by one of your integration activities that would be too expensive.

Base Offering

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

- Maximum of 100,000 records and 3 hours of execution per extraction transaction.
- Maximum of 250,000 extracted records per day.
- Maximum of 256 MB of data per extraction process.
- Maximum of 25,000 imported records per day.
- No limit on the length of time per import transaction.
- Maximum of 2 concurrent integration requests per customer per product. When there are more than 2 concurrent integration requests per customer, all subsequent requests will be queued and run later when there is an available slot to execute them.

Premium Offering

Customers with the premium offering are subject to higher maximum extracted and imported records per day limits.

The premium offering enables Customers to run data extracts from a Taleo production zone to their own data warehouse with a higher volume limit. Data warehouse to be provided by Customer. The premium offering does not include production support services, integration script maintenance or consulting services related to specific integrations established for Data Warehouse Extraction.

Level 1

- Maximum of 100,000 records and 3 hours of execution per extraction transaction.
- Maximum of 1,500,000 extracted records per day.
- Maximum of 256 MB of data per extraction process.
- Maximum of 75,000 imported records per day.
- No limit on the length of time per import transaction.
- Maximum of 2 concurrent integration requests per customer per product. When there are more than 2 concurrent integration requests per customer, all subsequent requests will be queued and run later when there is an available slot to execute them.

Level 2
Appendices

Appendix A

- Maximum of 200,000 records and 3 hours of execution per extraction transaction.
- Maximum of 5 million extracted records per day.
- Maximum of 512 MB of data per extraction process.
- Maximum of 150,000 imported records per day.
- No limit on the length of time per import transaction.
- Maximum of 2 concurrent integration requests per customer per product. When there are more than 2 concurrent integration requests per customer, all subsequent requests will be queued and run later when there is an available slot to execute them.

Level 3

- Maximum of 300,000 records and 3 hours of execution per extraction transaction.
- Maximum of 10 million extracted records per day.
- Maximum of 512 MB of data per extraction process.
- Maximum of 150,000 imported records per day. Can go up to 350,000 records per day for exceptional circumstances.*
- No limit on the length of time per import transaction.
- Maximum of 2 concurrent integration requests per customer per product. When there are more than 2 concurrent integration requests per customer, all subsequent requests will be queued and run later when there is an available slot to execute them.

Planned Exception Framework

In the situation 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 project types with corresponding volume limits.

<table>
<thead>
<tr>
<th>Project Type</th>
<th>Daily Import</th>
<th>Daily Export</th>
<th>Per Transaction Export</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Migrations (New / V2T) – No OLTP users</td>
<td>350,000*</td>
<td>250,000</td>
<td>100,000, 256 MB</td>
</tr>
<tr>
<td>Data Migrations (Acquisitions / Expansion) – OLTP users</td>
<td>150,000*</td>
<td>250,000</td>
<td>100,000, 256 MB</td>
</tr>
<tr>
<td>Data warehouse initial extract – OLTP users</td>
<td>150,000*</td>
<td>10 million</td>
<td>100,000, 256 MB</td>
</tr>
<tr>
<td>Final Extract (non- Oracle dump)</td>
<td>150,000*</td>
<td>10 million</td>
<td>100,000, 256 MB</td>
</tr>
<tr>
<td>Large scale data refresh or modifications**</td>
<td>150,000*</td>
<td>250,000</td>
<td>100,000, 512 MB</td>
</tr>
</tbody>
</table>

* 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 customer 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 total per year maybe requested under this category of project type per year.

Planned Exception Process

Submit a Support ticket at minimum of 10 days in advance of the planned exception.
The content of this ticket must contain the following text with the appropriate answers describing the volume limits modification.

- Incident Information:
- Request Type: Configuration
- CS Priority: High
- Change Request: Setting Change / Activation
- Product: Taleo Connect
- Component: Taleo Connect Client or Taleo Connect Broker
- Zone: Production (only)
- Zone URL: xxxx.taleo.net
- Title: Request for modification for Volume Limits
- Description:
  - When are the volume limit modifications needed to be in place, please specify the to & from dates in the following format. YYYY-MM-DDThh:mm:ss-TZD
  - Please specify the reason for this request (i.e. Initial load Special event…)

The volume limit request should be one of the previous Project Type Confirmation will be made via the Support ticket.

For additional oversight, customers may wish to engage their Client Executive in the notification process for the request for a planned exception, though this is not required.

**Unplanned Exception Process**

Submit a Support ticket at the point that integration is identified as being the cause of the business impact on a customer’s zone.

The content of this ticket must contain the following text with the appropriate answers describing the volume limits modification.

- Incident Information:
- Request Type: Configuration
- CS Priority: Business Stand
- Change Request: Setting Change / Activation
- Product: Taleo Connect
- Component: Taleo Connect Client or Taleo Connect Broker
- Zone: Production (only)
- Zone URL: xxxx.taleo.net
- Title: URGENT request for modification for Volume Limits
- Description:
  - Please specify the reason for this request (i.e. Initial load Special event…)

The unplanned exception volume limit request will be reviewed. Confirmation will be made via the Support ticket.

For additional oversight, customer should inform their Client Executive of the escalation if possible, though this is not required.

The key difference is that this process is driven by a business stand CS Priority.
Appendix B

TCC Errors

TCC Errors usually occur during log in and return an error code.

302
Problem: Protocol HTTP and Port 80 is not a valid combination.
Solution: Select HTTPS and port 443.

401
Problem: the User Name and/or Password and/or ZoneName is invalid, and/or the User Role is different than System Integration
Solution: Validate that you can manually log in the Taleo Administrator WebTop with this User, verify the User Role, make corrections to your TCC environment file if required, and log in to TCC again

402
Problem: the Taleo zone in currently in maintenance, e.g. Taleo is performing an update or migration (ex: SP, EP, version)
Solution: You have to wait until the Taleo update or migration is completed

403
Problem: Forbidden Access, User does not have the right privileges to access this product, can also happen in desynchronisation.
Solution: Verify that user name, password, role, and communication protocols are valid for connection to Taleo zone.

404
Problem: You have selected a version of the data model in TCC (ex: 7.5) that is higher than your Taleo zone (ex: 7.0)
Solution: Close TCC, and log again using the data model version that corresponds to your Taleo zone

407
Problem: There is a Proxy authentication error
Solution: Verify your taleoconnectclient.ini file with the proxy information (see User Guide)

500
Problem: Internal server error, zone received document but failed to process it, usually because document is invalid
Solution: Run the TCC import/export again

The Taleo product could not be accessed
- Problem: the ZoneName is invalid (ex: stagingzone750.taleo.net)
Solution: Make corrections to your TCC connectivity information and log in to TCC again.

- Problem: Protocol HTTP and Port 443 is not a valid combination.
  Solution: Select HTTPS and port 443.

The entity is already booked

- The entity record to update is currently being edited by a User. After a certain delay this error message is returned since the record can’t be updated.

  For example:

  "Index","Identifier","Status","TransactionType","Result","Message"
  "1","atemplate001","error","jobtemplate.merge","The entity is already booked.;"

Out of Memory

- Java heap size memory management parameter -Xmx<maximum heap size> has been added to the taleoconnectclient.bat and taleoconnectclient.sh scripts. If out of memory errors occur when launching TCC from the command line, increase the parameter value or add the parameter if it’s missing.
Appendix C

Compatibility

TCC integration point compatibility with Taleo products is guaranteed one version up, as long as the reference model does not change. Compatibility for a TCC integration point is not guaranteed when the Taleo product reference model changes.

Architecture

Taleo Connect Client (TCC) communicates with Taleo Connect Server (TCS) that is embedded in every Taleo product. TCS uses internal reference models that describe each Taleo product entity available for integration. The reference models are related to the version of the business engine; the business engine contains sets of business rules that integration must follow.

The reference models are not automatically compatible with each other. Taleo products are enhanced and modified from one version to the next. The business rules change to comply with the enhancements and modifications.

Example

An integration point built with reference model 7.0 for a version 7.0.X product will continue to function after the product is upgraded to version 7.5.X, as long as the reference model remains unchanged. In this case, compatibility is guaranteed.

Product and model compatibility

If the reference model is modified, to take advantage of a new field for example, then compatibility is not guaranteed. In this case, the integration points must be re-tested and modified if necessary.

A modification to a business rule may affect integration in a minimal way. Testing is required.

Note: The Recruiting/Professional and SmartOrg models have changed considerably from 7.5 to 10, please refer to the Taleo Integration Migration Guide for details.

Migration

Migration of integration points to the current version of the product model is vital to maintaining future compatibility.

When the Taleo product is upgraded to a new version, the TCC feature packs should be upgraded to the same version and the integration points migrated to the corresponding business model.

Migration Process
1. Integration points are created for TEE 7.0 using reference model 7.0.
2. The Taleo product is upgraded to TEE 7.5. It offers connect model 7.5 and also model 7.0 to maintain backwards compatibility. The model 7.0 integration points continue to function.
3. TCC is upgraded to version 7.5. The integration points are migrated to reference model 7.5.
4. The Taleo product is upgraded to Taleo Enterprise 10.0. It offers connect model 10.0 and also model 7.5 to maintain backwards compatibility. The model 7.5 integration points that are compatible continue to function. If the integration points are not migrated and still use reference model 7.0, they fail to function.
5. TCC feature packs are upgraded to the latest available version. The integration points are migrated to reference model 10.0.

TCC comes with a migration helper tool to easily migrate import and export specifications from a specific model to another. The migration tool will open a new import or export editor window and try to convert the integration point from one model to the next. The new file is named: *.migration*.xml.

During migration, some fields may not be migrated because they no longer exist or are not available in the new model. When this happens, a Migration Warning appears in the report indicating that a manual migration must be done to complete the process. The fields that must be deleted or replaced are clearly identified.

For additional details, refer to the Taleo Integration Migration Guide.

HTML Migration Report
Appendices

Taleo Connect Client

Migration Warning
Manual migration must be done before the next execution.

Taleo Connect Client
Queryfile migration report

General information

<table>
<thead>
<tr>
<th>Type</th>
<th>Product Code</th>
<th>Model Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>From Query</td>
<td>PRO704</td>
<td><a href="http://www.taleo.com/ws/art7/001/2006/12">http://www.taleo.com/ws/art7/001/2006/12</a></td>
</tr>
<tr>
<td>To Query</td>
<td>PRO755</td>
<td><a href="http://www.taleo.com/ws/art755/2006/12">http://www.taleo.com/ws/art755/2006/12</a></td>
</tr>
</tbody>
</table>

Migration information

<table>
<thead>
<tr>
<th>Root Entity</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>Candidate</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Status</th>
<th>Comment</th>
<th>Rule</th>
<th>Entity</th>
<th>Motive</th>
<th>Operation</th>
<th>Operation Detail</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Compatibility</th>
<th>Details</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Path</th>
<th>Status</th>
<th>Comment</th>
<th>Rule</th>
<th>Entity</th>
<th>Field / Relation</th>
<th>Motive</th>
<th>Operation</th>
<th>Operation Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credentials-Agent.Number</td>
<td>WARNING</td>
<td>The field Entity.Number has been set hidden</td>
<td>Entity</td>
<td>Number</td>
<td>The Number field is an internal key that was incorrectly exposed in the previous version</td>
<td>Field Scope Hidden</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Extra information

Messages